STATISTICAL METHODS FOR FUNCTION ESTIMATION AND CLASSIFICATION

A Thesis
Presented to
The Academic Faculty

by

Heeyoung Kim

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology
August 2011

STATISTICAL METHODS FOR FUNCTION ESTIMATION AND CLASSIFICATION

Approved by:

Professor Xiaoming Huo, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Roshan Joseph Vengazhiyil
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Jianjun Shi
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Liang Peng
School of Mathematics
Georgia Institute of Technology

Professor Jye-Chyi Lu
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: 22 April 2011

To my parents and husband

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Dr. Xiaoming Huo, for his guidance and support during my doctoral study. Without his help and encouragement, the accomplishment of this dissertation would not be possible. I am also very grateful to Dr. Jianjun Shi, Dr. Jye-Chyi Lu, Dr. Roshan Joseph Vengazhiyil, and Dr. Liang Peng for serving on my committee and giving me insightful comments. I would like to especially thank my mother, father, sisters, and brother for their love and support. Finally, my heartfelt gratitude goes to my husband, Sungil, who has been a tremendous source of encouragement throughout the past five years. His support and love have made this endeavor possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis consists of three chapters. Chapter 1 focuses on adaptive smoothing splines for fitting functions with varying roughness. Smoothing splines are widely used for estimating an unknown function in the nonparametric regression. If data have large spatial variations, however, the standard smoothing splines (which adopts a global smoothing parameter λ) performs poorly. Adaptive smoothing splines adopt a variable smoothing parameter $\lambda(x)$ (i.e., the smoothing parameter is a function of the design variable x) to adapt to varying roughness. Here the choice of $\lambda(x)$ is important, since it controls the trade-off between smoothness of the estimated function and the goodness of fit. In Chapter 1, we study the asymptotically optimal choice of $\lambda(x)$ in adaptive smoothing splines. Chapter 1 consists of two parts.

In the first part of Chapter 1, we study an asymptotically optimal procedure to choose a multivariate smoothing parameter (discretized version of $\lambda(x)$) in adaptive smoothing splines. We first derive a closed-form adaptive smoothing spline estimator which involves the multivariate smoothing parameter. We show that the derived estimator achieves the optimal rate of convergence. We then propose to choose the multivariate smoothing parameter by minimizing a multivariate version of the generalized cross validation (mGCV) function. With the choice by mGCV, the resulting estimator is shown to be consistent and asymptotically optimal under some general conditions. Numerical examples demonstrate the advantage of our method against ordinary smoothing splines, as well as other contemporary adaptive splines methods.

In the second part of Chapter 1, we derive the asymptotically optimal local penalty function under suitable conditions. The derived locally optimal $\lambda(x)$ is subsequently used for the derivation of the locally optimal smoothing spline estimator. For the

implementation of the proposed method, an iterative adaptive scheme is proposed. In the simulation study, our method is compared with other contemporary smoothing spline methods that consider spatial adaptation, and shows its outstanding performance in finite sample cases.

In Chapter 2, we propose a Lipschitz regularity based statistical model, and apply it to coordinate measuring machine (CMM) data to estimate the form error of a manufactured product and to determine the optimal sampling positions of CMM measurements. In contemporary manufacturing, coordinate measuring machines (CMMs) are widely used for dimensional inspection. To achieve accurate inspection while minimizing the cost and time, we must address the following two critical issues: (1) How do we select the positions of the sampling points so that we can get as much information from a limited number of samples as possible; and (2) given the limited number of measurement points, how do we construct condence bands for the examined geometric feature and assess the form error so that one can reliably decide whether the product is acceptable? We propose a wavelet-based model that takes advantage of the fact that the Lipschitz regularity holds for the surface. Based on the wavelet framework, the proposed model yields the optimal sampling positions. After obtaining the measurements, one can then construct confidence bands and estimate the form error using the proposed wavelet-based random curve interpolating scheme. The proposed method is validated using both synthetic and real CMM data concerning straightness. The comparison with other existing methods demonstrates the effectiveness of our method.

Chapter 3 focuses on the classification of functional data which are known to be well separable within a particular interval. In many applications, it is desirable to build a classifier that is bounded within an interval. Our motivating example is rooted in monitoring in a stamping process. A novel approach is proposed and examined. Our method consists of three stages: (1) A baseline of each class is estimated via

convex optimization; (2) an "optimal interval" that maximizes the difference among the baselines is identified; and (3) a classifier that is based on the "optimal interval" is constructed. We analyze the implementation strategy and properties of the derived algorithm. The derived classifier is named an interval based classifier (IBC) and can be computed via a low-order-of-complexity algorithm. Comparing to existing state-of-the-art classifiers, we illustrate the advantages of our approach. To showcase its usage in applications, we apply the IBC to a set of tonnage curves from stamping processes, and observe superior performance. The proposed method can help identifying fault conditions in manufacturing.

CHAPTER I

ASYMPTOTICALLY OPTIMAL ADAPTIVE SMOOTHING SPLINES

1.1 Introduction

We consider the problem of estimating an unknown function $f(\cdot)$ given observations

$$y_i = f(x_i) + \epsilon_i, \quad i = 1, \ldots, n, \tag{1.1}$$

where the design points x_i follow a strictly positive continuous density function and ϵ_i are random noise with mean 0 and unknown variance σ^2. Such a formulation is commonly used in nonparametric functional estimation. Without loss of generality, we assume that the domain of f is $[0, 1]$. We also assume that data are available on both sides beyond 0 and 1 to avoid consideration of boundary effects. Smoothing spline is one of the most popular methods for estimating f. Let $f^{(m)}$ denote the mth derivative of f. The smoothing spline estimators are the unique solutions of the following problem

$$\min_{f \in W_2^m[0,1]} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int_0^1 \{f^{(m)}(x)\}^2 dx, \tag{1.2}$$

where $f^{(m)}$ is the mth derivative of f, $W_2^m[0, 1]$ is the mth order Sobolev space defined as $\{f : f^{(j)}$ is absolutely continuous for $j = 0, \ldots, m-1,$ and $f^{(m)} \in L_2[0, 1]\}$, and λ is a smoothing parameter which controls the trade-off between smoothness of the estimated function (the second term) and the goodness of fit (the first term). Large values of λ produce smoother functions while smaller values produce more wiggly functions. For the automatic choice of λ, many procedures have been proposed, including cross validation (CV) [47], generalized cross validation (GCV) [9], and Mallow's C_p [35].

Even though smoothing splines have been shown to perform well in many examples, if the underlying function is spatially nonhomogeneous, traditional smoothing splines will fail to capture the varying degrees of smoothness properly. In practice, there are various types of functions with varying smoothness, and two scenarios [13] are illustrated in Fig. 1. The functions in Fig. 1 change rapidly in some regions while

(a) Doppler function (b) Bumps function

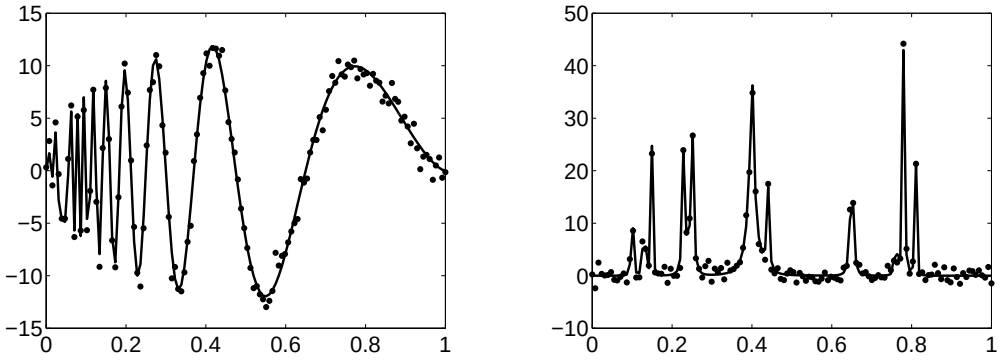


Figure 1: Illustration of two widely-studied functions with varying degrees of smoothness: (a) Doppler function, (b) Bumps function. Solid curve is the true function and dots are noisy observations.

being smooth in others. If we choose the global smoothing parameter to be relatively small, the resulting spline estimate will describe the function well in the regions of large variations, however it will under-smooth in other regions. On the other hand, if the global smoothing parameter is chosen to be relatively large, then the estimated function will be over-smoothed in the regions of large variations. This indicates that in fitting functions with varying roughness, using a global smoothing parameter is not sufficient.

To resolve such a problem and attain more flexible estimation of the function, there have been attempts to allow for the smoothing parameter to vary adaptively with x [2, 38, 48, 33]. Instead of (1.2), the following minimization problem has been considered:

$$\min_{f \in W_2^m[0,1]} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \int_0^1 \lambda(x) \{f^{(m)}(x)\}^2 dx, \tag{1.3}$$

where $\lambda(x)$ is a *variable* smoothing parameter—a function of x. The framework in (1.3) is just one of many ways to achieve spatial adaptivity. Other approaches include, e.g., multiscale analysis (like wavelets), orthogonal basis functions (like Fourier transform), kernel or local polynomial regression with variable bandwidth, and more. In this chapter, we focus on the smoothing spline framework in (1.3), since it gives a natural way to study how adjusted penalization can improve the adaptivity of the derived estimator. In addition, certain asymptotic properties, like the asymptotic optimality that will be established in this chapter, are not known under other frameworks.

In the framework given by (1.3), there have been two major approaches. The first approach assumes that $\lambda(x)$ is a step function, i.e., a piecewise constant function. [38] assumed an equal-size piecewise structure for $\lambda(x)$. The number of jumps and the jump locations need to be prespecified. [33] extended the work of [38]. They also assume a step function for $\lambda(x)$, but the segmentation is data driven. The number of jumps and the jump locations are chosen based on the data. The step function is estimated by maximizing the generalized likelihood. Although this work improves the flexibility of the segmentation of $\lambda(x)$ over those by [38], it still assumes that the smoothness of the function is constant in-between the small number of jump points, and changes abruptly at the jump points. Moreover, the optimal choice of the smoothing parameter and the associated asymptotic properties have not been studied in the literature.

The second approach assumes a particular continuously varying penalty function for $\lambda(x)$. [2] assumed that $\lambda(x)$ is proportional to $(f^{(2)}(x))^{-2}$. [48] considered one more tuning parameter γ, and assumed that $\lambda(x)$ is proportional to $(f^{(2)}(x))^{-2\gamma}$. They recommended to choose the value of γ via CV among the options of $\gamma = 1, 2, 4$. In these works, once the estimate of $(f^{(2)}(x))^{-2}$ or $(f^{(2)}(x))^{-2\gamma}$ is obtained, $\lambda(x)$ is rewritten as $\tau(\tilde{f}^{(2)}(x))^{-2}$ or $\tau(\tilde{f}^{(2)}(x))^{-2\gamma}$, where $(\tilde{f}^{(2)}(x))^{-2}$ and $(\tilde{f}^{(2)}(x))^{-2\gamma}$ denote the estimate of the $(f^{(2)}(x))^{-2}$ and $(f^{(2)}(x))^{-2\gamma}$, respectively. Then only one global

smoothing parameter τ remains to be determined, and it is chosen via CV or GCV. These methods are computationally efficient, since only one smoothing parameter needs to be chosen. However, the assumptions of $(f^{(2)}(x))^{-2}$ or $(f^{(2)}(x))^{-2\gamma}$ have been made somewhat intuitively, rather than based on some theoretical justification.

The first and second parts of this chapter are in line with the above first and second approaches, respectively. In the first part (Section 1.2), we study an asymptotically optimal procedure to choose the multivariate smoothing parameter under the assumption of the step function for $\lambda(x)$. In the second part (Section 1.3), we derive the asymptotically optimal local penalty function of $\lambda(x)$.

1.2 Asymptotic Optimality of the Generalized Cross Validation in Adaptive Smoothing Splines

1.2.1 Introduction

We study an optimal choice for $\lambda(x)$ and propose an asymptotically optimal smoothing spline estimator for fitting functions with varying smoothness. We consider a reasonable discretized version of (1.3): we assume that $\lambda(x) \equiv \lambda_i$ for $x \in [x_i, x_{i+1}), 1 \leq i \leq n-1$ and $x_1 < x_2 < \ldots, x_n$. Then we need to estimate a multivariate smoothing parameter $\boldsymbol{\lambda}$ whose dimension is the sample size minus one (the ith component of $\boldsymbol{\lambda}$ is λ_i). Under this framework, we derive the closed-form solution and we show that this estimator achieves the optimal rate of convergence in terms of the mean squared error. For the optimal choice of $\boldsymbol{\lambda}$, we propose to use a multivariate version of the generalized cross validation (mGCV). We show that if we choose $\boldsymbol{\lambda}$ by minimizing the mGCV, then the resulting estimate is *consistent* in the sense that the true loss tends to zero as the sample size goes to infinity, and is *asymptotically optimal* in the sense that it achieves the smallest possible loss in probability when the sample size goes to infinity. We show that our theoretical results still hold for the special case when $\lambda(x)$ is a step function with the number of jumps much less than the sample size. In addition to nice asymptotic properties, the proposed estimator with mGCV choice

of $\boldsymbol{\lambda}$ also preforms well in finite sample cases, which will be shown in our simulation study.

In approximating $\lambda(x)$ with a multivariate smoothing parameter $\boldsymbol{\lambda}$, we allow $\boldsymbol{\lambda}$ as flexible as possible by considering the aforementioned discretization. While the existing literature typically assumes that $\lambda(x)$ is piecewise constant or proportional to $(f^{(2)}(x))^{-2}$, this chapter does not have such a restriction on $\lambda(x)$, and chooses the optimal value of $\boldsymbol{\lambda}$ via a data-driven approach—minimizing the mGCV function. For the ordinary smoothing splines (i.e., with univariate λ), the optimal choice of λ has been extensively studied. In particular, it has been shown that if one chooses λ via GCV, the resulting estimate has nice asymptotic properties [9, 30, 31]. However, for the adaptive smoothing spline, similar study on the optimal choice of $\lambda(x)$ (or its discretized version $\boldsymbol{\lambda}$) has not appeared. In this chapter, we show that with some moderate conditions on $\boldsymbol{\lambda}$, the adaptive smoothing spline estimator with the mGCV choice of $\boldsymbol{\lambda}$ has the same asymptotic properties as the ones established for the ordinary smoothing splines—consistency and asymptotic optimality.

Note that the asymptotic optimality that will be established for our estimator is equivalent to saying that the proposed estimate is asymptotically the most efficient one among a class of estimators. Due to the flexibility of $\boldsymbol{\lambda}$, such a class (of estimators) is broad. For example, we can consider another class of estimators by assuming that $\lambda(x)$ is approximated with a step function with a small number of steps, such as in [38] and [33]. Since this is a special case of our general framework (note that we assume a step function with steps at the design points), our established asymptotic optimality still holds for this case: i.e., if we choose the constants in-between the steps by minimizing the mGCV, the resulting estimate is asymptotically optimal; the only difference is that the optimality holds over a different functional class. In summary, our theoretical results cover all possible cases of step functions assumed for $\lambda(x)$ with flexible step size and step number.

We must admit that the multivariate version of GCV is hard to minimize numerically, due to its non-convexity. However, in our finite sample simulations, a simple heuristic updating scheme can already render better performance in comparison to other contemporary methods. This demonstrates the potential of such an approach. More sophisticated numerical solutions will be a future research problem and will not be covered here.

The rest of this section is organized as follows. In Section 1.2.2, we review ordinary smoothing splines and GCV. We then introduce a new penalization method for spatial adaptation in Section 1.2.3. The key idea is to turn a univariate λ in Section 1.2.2 to a function of the location $\lambda(x)$, which is subsequently discretized to a multivariate $\boldsymbol{\lambda}$. Throughout this chapter, we use bold faced letters to denote the multivariate counterpart of the univariate parameter; e.g., $\boldsymbol{\lambda}$ is the multivariate counterpart of the univariate λ. Under very general conditions on $\boldsymbol{\lambda}$, the optimal rate of convergence for the derived estimator is shown in Section 1.2.4. For the choice of $\boldsymbol{\lambda}$, a multivariate version of the GCV (mGCV) is suggested in Section 1.2.5. The consistency and asymptotic optimality of our estimator with the mGCV choice of $\boldsymbol{\lambda}$ are shown in Section 1.2.6 and Section 1.2.7, respectively. In Section 1.2.8, we apply our method to several examples and show the effectiveness of the proposed method. Finally, we conclude in Section 1.2.9.

1.2.2 Reviews for Ordinary Smoothing Splines and Generalized Cross Validation

We briefly review ordinary smoothing splines. For more details, we refer to [21, 16]. Throughout this section, we focus on the cubic smoothing splines (i.e., with $m = 2$ in (1.2)) which are the most commonly used splines in practice. Using the cubic smoothing splines, f is estimated by minimizing the following objective function:

$$J(\lambda; f) = \sum_{i=1}^{n} [y_i - f(x_i)]^2 + \lambda \int_0^1 \{f^{(2)}(x)\}^2 dx. \tag{2.4}$$

Let $\mathbf{y} = (y_1, \ldots, y_n)^T$, $\mathbf{f} = (f(x_1), \ldots, f(x_n))^T$, $\delta_i = f^{(2)}(x_i), i = 2, \cdots, n-1$, $\delta_1 = \delta_n = 0$, and $h = x_{i+1} - x_i, 1 \leq i \leq n-1$. (The last equation indicates that we restrict ourselves to the case of equally spaced samples—more general case is approachable, however not described here.) Using these notations, the objective function in (2.4) can be restated as

$$J(\lambda; \mathbf{f}) \quad = \quad (\mathbf{y} - \mathbf{f})^T(\mathbf{y} - \mathbf{f}) + \lambda \delta^T \mathbf{M} \delta, \tag{2.5}$$

where $\delta = (\delta_2, \ldots, \delta_{n-1})^T$ and $\mathbf{M}$ is defined to be the $(n-2) \times (n-2)$ matrix with elements m_{ij}, given by $m_{ii} = \frac{2h}{3}$ for $i = 1, \cdots, n-2$; $m_{i,i+1} = m_{i+1,i} = \frac{h}{6}$ for $i = 1, \cdots, n-3$; zeros elsewhere. Using the fact that $\mathbf{M}\delta = \mathbf{Q}\mathbf{f}$, one can find the minimizer of $J(\lambda; \mathbf{f})$ as follows:

$$\hat{\mathbf{f}}(\lambda) = \left(\mathbf{I} + \lambda \mathbf{Q}^T \mathbf{M}^{-1} \mathbf{Q}\right)^{-1} \mathbf{y}, \tag{2.6}$$

where $\mathbf{Q}$ is defined to be the $(n-2) \times n$ matrix with elements q_{ij}, given by $q_{ii} = q_{i,i+2} = \frac{1}{h}$ and $q_{i,i+1} = \frac{-2}{h}$ for $i = 1, \cdots, n-2$, and zeros elsewhere.

In (2.6), the choice of λ is an important issue, and many procedures have been proposed for the optimal choice of λ. One of the most popular procedures is the generalized cross validation (GCV) [9]. Let $\mathbf{A}(\lambda) = \left(\mathbf{I} + \lambda \mathbf{Q}^T \mathbf{M}^{-1} \mathbf{Q}\right)^{-1}$. GCV selects λ by minimizing

$$\mathrm{GCV}_n(\lambda) = \frac{n^{-1} \|\mathbf{y} - \hat{\mathbf{f}}(\lambda)\|^2}{[n^{-1} \mathrm{tr}(\mathbf{I} - \mathbf{A}(\lambda))]^2}, \tag{2.7}$$

where $\mathbf{A}(\lambda)$ is called a smoothing matrix that satisfies $\hat{\mathbf{f}}(\lambda) = \mathbf{A}(\lambda)\mathbf{y}$, and 'tr' denotes trace. In the remainder of this subsection, we justify the adoption of GCV: if we choose λ by minimizing GCV, the resulting estimate minimizes the true loss for estimating f with $\hat{f}(\lambda)$ defined by

$$L_n(\lambda) \quad = \quad n^{-1} \|\mathbf{f} - \hat{\mathbf{f}}(\lambda)\|^2, \tag{2.8}$$

where $\|\cdot\|$ indicates the Euclidean norm and we use this norm throughout the chapter.

In the following, we use $\mathbf{A}_n(\lambda)$ (instead of $\mathbf{A}(\lambda)$) to integrate the sample size n. Similarly, $\mathbf{f}_n$ denote $\mathbf{f}$ when the sample size is n.

Theorem 1.2.1 (Li(1985)) *Consider the following Stein's estimate $\tilde{\mathbf{f}}_n(\lambda)$, the associated Stein's unbiased risk estimator ($SURE_n(\lambda)$), and the loss $\tilde{L}_n(\lambda)$ while estimating $\mathbf{f}_n$ by $\tilde{\mathbf{f}}_n(\lambda)$:*

$$\tilde{\mathbf{f}}_n(\lambda) = \mathbf{y}_n - \sigma^2 \frac{tr(\mathbf{I} - \mathbf{A}_n(\lambda))}{\|(\mathbf{I} - \mathbf{A}_n(\lambda))\mathbf{y}_n\|^2}(\mathbf{I} - \mathbf{A}_n(\lambda))\mathbf{y}_n,$$

$$SURE_n(\lambda) = \sigma^2 - \sigma^4 \frac{[n^{-1}tr(\mathbf{I} - \mathbf{A}_n(\lambda))]^2}{n^{-1}\|(\mathbf{I} - \mathbf{A}_n(\lambda))\mathbf{y}_n\|^2},$$

$$\tilde{L}_n(\lambda) = n^{-1}\|\tilde{\mathbf{f}}_n(\lambda) - \mathbf{f}_n\|^2.$$

Under the following conditions:

> ***(C.1)*** *The 4th moment of $\epsilon_i's$ are upper bounded by a constant, where ϵ_i are random noise in* (2.41),
>
> ***(C.2)*** *There exists a constant K, such that $\forall a > 0$,*
>
> $$\sup_{x \in \mathbb{R}} P\{x - a \leq \epsilon_i \leq x + a\} \leq K \cdot a, \quad \text{for } \forall i,$$

for any $\delta > 0$, we have

$$\sup_{\mathbf{f}_n \in \mathbb{R}^n} P\left\{ \sup_{\lambda \in \mathbb{R}^+} \left| SURE_n(\lambda) - \tilde{L}_n(\lambda) \right| > \delta \right\} \to 0.$$

Theorem 1.2.1 demonstrates that $SURE_n(\lambda)$ is a uniformly consistent estimator of $\tilde{L}_n(\lambda)$. Also note that minimizing the GCV function in (2.7) is equivalent to minimizing the $SURE_n(\lambda)$.

Remark 1.2.2 *The conditions **(C.1)** and **(C.2)** can be replaced by the following condition **(C.3)**:*

> ***(C.3)*** *ϵ_i are i.i.d. $N(0, \sigma^2)$. That is, **(C.3)** implies **(C.1)** and **(C.2)**.*

The asymptotic equivalence between $\tilde{\mathbf{f}}_n(\lambda)$ and $\hat{\mathbf{f}}_n(\lambda)$ are known as in the following theorem due to [31].

Theorem 1.2.3 (Li(1986)) *For any sequence λ_n such that $\hat{\mathbf{f}}_n(\lambda_n)$ is consistent in the sense that*

$$n^{-1}\|\mathbf{f}_n - \hat{\mathbf{f}}_n(\lambda_n)\|^2 \to 0, \tag{2.9}$$

under the condition that

$$(n^{-1}tr\mathbf{A}_n(\lambda_n))^2/n^{-1}tr\mathbf{A}_n^2(\lambda_n) \to 0, \tag{2.10}$$

and

$$\inf_{\lambda\geq 0} \quad n\cdot\mathbb{E}L_n(\lambda) \to \infty, \tag{2.11}$$

$\tilde{\mathbf{f}}_n(\lambda_n)$ and $\hat{\mathbf{f}}_n(\lambda_n)$ are asymptotically indistinguishable in the sense that

$$n^{-1}\|\tilde{\mathbf{f}}_n(\lambda_n) - \hat{\mathbf{f}}_n(\lambda_n)\|^2/L_n(\lambda_n) \to 0.$$

Let $\hat{\lambda}_G$ denote the value of λ chosen by minimizing GCV in (2.7). It is known [31] that if f is not a polynomial of degree 2 or less, (2.11) holds, and that if x_i are equispaced and ϵ_i are i.i.d. $N(0,\sigma^2)$, (2.9) and (2.10) hold with $\lambda_n = \hat{\lambda}_G$. Under these general conditions, therefore, $\tilde{\mathbf{f}}_n(\hat{\lambda}_G)$ and $\hat{\mathbf{f}}_n(\hat{\lambda}_G)$ are asymptotically equivalent due to Theorem 1.2.3. Together with Theorem 1.2.1, this demonstrates that if we choose λ by minimizing GCV, the resulting smoothing spline estimate $\hat{\mathbf{f}}_n(\hat{\lambda}_G)$ minimizes the true loss $L_n(\lambda)$ in (2.8).

1.2.3 A Strategy to Achieve Spatial Adaptivity

The essence of achieving *spatial adaptivity* is to utilize a $\lambda(x)$, which is a function of location x, instead of a constant λ. We assume $\lambda(x) \equiv \lambda_i$ for $x \in [x_i, x_{i+1})$, $1 \leq i \leq n-1$. (Recall we adopted the following: $x_1 < x_2 < \cdots < x_n$.) It is reasonable to assume that $\lambda(x)$ is constant in between neighboring observations, because we do not have information to determine its value in a finer resolution. Note that this is

more general than assuming a set of jump points distinct from the design points, such as the approaches in [38] and [33]. Due to this generality, our theoretical results established in this paper can be easily extended to the cases in [38] and [33].

Denote $\boldsymbol{\lambda}$ as a vector: $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{n-1})$. We then can rewrite the function in (1.3) with $m = 2$ as

$$J^\star(\boldsymbol{\lambda}; \mathbf{f}) = (\mathbf{y} - \mathbf{f})^T(\mathbf{y} - \mathbf{f}) + \frac{h}{3}\delta^T \mathbf{M}(\boldsymbol{\lambda})\delta, \tag{2.12}$$

where δ was defined previously and $\mathbf{M}(\boldsymbol{\lambda}) \in \mathbb{R}^{(n-2)\times(n-2)}$ satisfies $\mathbf{M}(\boldsymbol{\lambda})_{ii} = \lambda_i + \lambda_{i+1}$, $1 \le i \le n-2$, $\mathbf{M}(\boldsymbol{\lambda})_{i,i+1} = \mathbf{M}(\boldsymbol{\lambda})_{i+1,i} = \frac{\lambda_{i+1}}{2}$, $1 \le i \le n-3$, and zeros elsewhere. The details of the derivation of (2.12) are in Appendix A.1. Using the fact that $\mathbf{M}\delta = \mathbf{Q}\mathbf{f}$ and by considering the first order condition, one can obtain the minimizer of (2.12), which has the following closed-form:

$$\hat{\mathbf{f}}(\boldsymbol{\lambda}) = \left(\mathbf{I} + \frac{h}{3}\mathbf{Q}^T\mathbf{M}^{-1}\mathbf{M}(\boldsymbol{\lambda})\mathbf{M}^{-1}\mathbf{Q}\right)^{-1}\mathbf{y}. \tag{2.13}$$

We will propose the optimal means to decide $\boldsymbol{\lambda}$ in (2.13) in Section 1.2.5.

1.2.4 Optimal Rate of Convergence

The optimal rate of convergence in nonparametric function estimation was derived in [46]. Since then, many methods have been proved to achieve this optimal rate. In the following theorem, we show that the proposed estimator in (2.13) also achieves the optimal rate of convergence, under a very general condition on $\boldsymbol{\lambda}$.

Theorem 1.2.4 *For the estimator* $\hat{\mathbf{f}}$ *in* (2.13)*, if both* $\min(\boldsymbol{\lambda}_n)$ *and* $\max(\boldsymbol{\lambda}_n)$ *are* $O_p(n^{-4/5})$ *where* $\min(\boldsymbol{\lambda}_n)$ *and* $\max(\boldsymbol{\lambda}_n)$ *denote the minimum and maximum of* λ_i*,* $i = 1, \cdots, n-1$*, then* $n^{-1}\|\mathbf{f} - \hat{\mathbf{f}}(\boldsymbol{\lambda})\|^2 = O_p(n^{-4/5})$.

The above theorem is provable via known results in empirical processes; an example proof can be found in Theorem 1 of [48]. Because such a proof is known by standard techniques, we omit it in this section. Note that $O_p(n^{-4/5})$ is the optimal nonparametric rate for $m = 2$. More general results (for arbitrary m) is possible. The above

theorem provides a hint on the conditions that one may want to impose on $\boldsymbol{\lambda}$. That is, the ratio $\max(\boldsymbol{\lambda}_n)/\min(\boldsymbol{\lambda}_n)$ should be upper bounded by a constant. We will see that such a condition is the only condition that we need to impose on $\boldsymbol{\lambda}$, in order to prove asymptotic properties of our proposed estimator.

1.2.5 Multivariate Version of GCV (mGCV)

For the proposed estimator in (2.13), the choice of the multivariate smoothing parameter $\boldsymbol{\lambda}$ is important for appropriate spatial adaptation. We suggest to use the *multivariate version of generalized cross validation (mGCV)* defined as

$$\mathrm{mGCV}_n(\boldsymbol{\lambda}) = \frac{n^{-1}\|\mathbf{y} - \hat{\mathbf{f}}(\boldsymbol{\lambda})\|^2}{[n^{-1}\mathrm{tr}(\mathbf{I} - \mathbf{S}(\boldsymbol{\lambda}))]^2}, \tag{2.14}$$

where $\mathbf{S}(\boldsymbol{\lambda}) = \left(\mathbf{I} + \frac{h}{3}\mathbf{Q}^T\mathbf{M}^{-1}\mathbf{M}(\boldsymbol{\lambda})\mathbf{M}^{-1}\mathbf{Q}\right)^{-1}$. Note that we have $\hat{\mathbf{f}}(\boldsymbol{\lambda}) = \mathbf{S}(\boldsymbol{\lambda})\mathbf{y}$.

Parallel with Theorem 1.2.1, we establish the uniform consistency of $SURE_n(\boldsymbol{\lambda})$ in the following theorem. Together with the fact that minimizing mGCV is equivalent to minimizing $SURE_n(\boldsymbol{\lambda})$, the following theorem gives a justification of the mGCV. In the following, we use $\mathbf{S}_n(\boldsymbol{\lambda})$ (instead of $\mathbf{S}(\boldsymbol{\lambda})$) to take into account the sample size n.

Theorem 1.2.5 *Consider the following Stein estimate $\tilde{\mathbf{f}}_n(\boldsymbol{\lambda})$, the associated unbiased risk estimate ($SURE_n(\boldsymbol{\lambda})$), and the true loss $\tilde{L}_n(\boldsymbol{\lambda})$ while estimating $\mathbf{f}_n$ by $\tilde{\mathbf{f}}_n(\boldsymbol{\lambda})$:*

$$\tilde{\mathbf{f}}_n(\boldsymbol{\lambda}) = \mathbf{y}_n - \sigma^2\frac{tr(\mathbf{I} - \mathbf{S}_n(\boldsymbol{\lambda}))}{\|(\mathbf{I} - \mathbf{S}_n(\boldsymbol{\lambda}))\mathbf{y}_n\|^2}(\mathbf{I} - \mathbf{S}_n(\boldsymbol{\lambda}))\mathbf{y}_n,$$

$$SURE_n(\boldsymbol{\lambda}) = \sigma^2 - \sigma^4\frac{[n^{-1}tr(\mathbf{I} - \mathbf{S}_n(\boldsymbol{\lambda}))]^2}{n^{-1}\|(\mathbf{I} - \mathbf{S}_n(\boldsymbol{\lambda}))\mathbf{y}_n\|^2}, \tag{2.15}$$

and

$$\tilde{L}_n(\boldsymbol{\lambda}) = n^{-1}\|\mathbf{f}_n - \tilde{\mathbf{f}}_n(\boldsymbol{\lambda})\|^2.$$

Then $SURE_n(\boldsymbol{\lambda})$ is a uniformly consistent estimate of $\tilde{L}_n(\boldsymbol{\lambda})$ over $\mathbf{f}_n$ and $\boldsymbol{\lambda}$: For any $\delta > 0$,

$$\sup_{\mathbf{f}_n \in \mathbb{R}^n} P\left\{\sup_{\boldsymbol{\lambda} \in \mathbb{R}^{n-1}} |SURE_n(\boldsymbol{\lambda}) - \tilde{L}_n(\boldsymbol{\lambda})| > \delta\right\} \to 0.$$

Proof of the above theorem is in Appendix A.2. Note that the Theorem 1.2.5 establishes the uniform consistency between $SURE_n(\boldsymbol{\lambda})$ and $\tilde{L}_n(\boldsymbol{\lambda})$ which involves $\tilde{\mathbf{f}}_n(\boldsymbol{\lambda})$. We would like to consider the original loss $L_n(\boldsymbol{\lambda})$ for estimating $\mathbf{f}_n$ with $\hat{\mathbf{f}}_n(\boldsymbol{\lambda})$. In the following, we establish the asymptotic equivalence between $\tilde{\mathbf{f}}_n(\boldsymbol{\lambda})$ and $\hat{\mathbf{f}}_n(\boldsymbol{\lambda})$.

Theorem 1.2.6 *For any $\hat{\boldsymbol{\lambda}}$ such that*

$$L_n(\hat{\boldsymbol{\lambda}}) \to 0, \tag{2.16}$$

and

$$\frac{(n^{-1}tr\mathbf{S}_n(\hat{\boldsymbol{\lambda}}))^2}{n^{-1}tr\mathbf{S}_n^2(\hat{\boldsymbol{\lambda}}))} \to 0, \tag{2.17}$$

under the following condition,

(A.1) $\inf_{\boldsymbol{\lambda}\in\mathbb{R}_+^{n-1}} \quad n \cdot \mathbb{E}L_n(\boldsymbol{\lambda}) \to \infty,$

we have

$$\frac{|SURE_n(\hat{\boldsymbol{\lambda}}) - \tilde{L}_n(\hat{\boldsymbol{\lambda}}) - n^{-1}\|\boldsymbol{\epsilon}_n\|^2 + \sigma^2|}{L_n(\hat{\boldsymbol{\lambda}})} \to 0, \tag{2.18}$$

and

$$\frac{n^{-1}\|\tilde{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}}) - \hat{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}})\|^2}{L_n(\hat{\boldsymbol{\lambda}})} \to 0. \tag{2.19}$$

Proof of the above theorem is in Appendix A.3. **(A.1)** states that the optimal rate of convergence of $\mathbb{E}L_n(\boldsymbol{\lambda})$ to zero must be slower than n^{-1}. For **(A.1)**, in the typical polynomial spline smoothing problems, $\inf_{\lambda>0}\mathbb{E}L_n(\lambda)$ tends to zero at the rate $n^{-1+\delta}$ for some small constant $\delta > 0$ except if the underlying function is the sampled values of a low order polynomial [54]. In our framework, we need to study when **(A.1)** holds—this is an open problem.

Let $\hat{\boldsymbol{\lambda}}_{mG}$ denote the value of $\boldsymbol{\lambda}$ chosen by minimizing the mGCV function in (2.14). We can show that under certain conditions, $\tilde{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}}_{mG})$ and $\hat{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}}_{mG})$ are asymptotically equivalent by showing that $\hat{\boldsymbol{\lambda}}_{mG}$ satisfies (2.16) and (2.17): (2.16) is essentially Theorem 1.2.9 that is to be established in the next subsection, and (2.17) is shown at the beginning of the proof of Theorem 1.2.11 in Appendix A.6.

1.2.6 Consistency of mGCV

We say that $\hat{\mathbf{f}}_n(\boldsymbol{\lambda})$ is consistent if $L_n(\boldsymbol{\lambda}) \to 0$ as $n \to \infty$, where $L_n(\boldsymbol{\lambda})$ is the loss while estimating $\mathbf{f}_n$ by $\hat{\mathbf{f}}_n(\boldsymbol{\lambda})$, i.e., $L_n(\boldsymbol{\lambda}) = n^{-1}\|\mathbf{f}_n - \hat{\mathbf{f}}_n(\boldsymbol{\lambda})\|^2$. In this section, we show that if we choose $\boldsymbol{\lambda}$ via mGCV, then the resulting $\hat{\mathbf{f}}$ is consistent. To establish the consistency of mGCV, we need the following two conditions:

(A.2) Recall $\mathbf{S}_n(\boldsymbol{\lambda}) = (\mathbf{I} + \boldsymbol{\Sigma}_n(\boldsymbol{\lambda}))^{-1}$, where $\boldsymbol{\Sigma}_n(\boldsymbol{\lambda}) = \frac{h}{3}\mathbf{Q}^T\mathbf{M}^{-1}\mathbf{M}(\boldsymbol{\lambda})\mathbf{M}^{-1}\mathbf{Q}$. Let $0 \le \tau_1 \le \tau_2 \le \cdots \le \tau_n$ be the eigenvalues of $\boldsymbol{\Sigma}_n(\boldsymbol{\lambda})$. For any m such that $m/n \to 0$, we have

$$\frac{\left(n^{-1}\sum_{i=m+1}^{n}\tau_i^{-1}\right)^2}{n^{-1}\sum_{i=m+1}^{n}\tau_i^{-2}} \to 0, \text{ as } n \to \infty,$$

(A.3) There exists $\boldsymbol{\lambda}_n$, such that $L_n(\boldsymbol{\lambda}_n) \to 0$.

The above **(A.2)** involves eigenvalues, and seems hard to verify. The following theorem provides a simple sufficient condition for it. Note such a condition is consistent with the discussion at the end of Section 1.2.4.

Theorem 1.2.7 *For the proposed estimator defined in* (2.13)*, if*

$$\max(\boldsymbol{\lambda}_n)/\min(\boldsymbol{\lambda}_n) < \textit{Constant, as } n \to \infty,$$

where $\max(\boldsymbol{\lambda}_n)$ *and* $\min(\boldsymbol{\lambda}_n)$ *denote the maximal and minimal values among* λ_i, $1 \le i \le n-1$, *then condition* ***(A.2)*** *holds.*

Proof of the above theorem is in Appendix A.4.

Remark 1.2.8 *It is known [19] that* ***(A.2)*** *can be replaced with the following weaker condition:*

(A.2′) *There exist constants* p *and* q, $0 < p < q < 1$ *such that* $\limsup \tau_{[pn]}/\tau_{[qn]} < 1$, *where* $[x]$ *denotes the greatest integer less than or equal to* x.

For **(A.3)**, it is known [30, Theorem 5.5] that such $\boldsymbol{\lambda}_n$ exists for the nonadaptive case if x_i's are equispaced and ϵ_i's are i.i.d. $N(0, \sigma^2)$. Since the nonadaptive smoothing spline is a special case of the spatially adaptive smoothing spline, under the same conditions, **(A.3)** holds for the spatially adaptive smoothing splines too.

The following theorem establishes the consistency of mGCV.

Theorem 1.2.9 (Consistency) *Under **(A.2)** and **(A.3)**, $\hat{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}}_{mG})$, where $\hat{\boldsymbol{\lambda}}_{mG}$ is the mGCV choice, is consistent, i.e., $L_n(\hat{\boldsymbol{\lambda}}_{mG}) \to 0$.*

Proof of the above theorem is in Appendix A.5. Based on Theorem 1.2.7 and the discussion on **(A.3)** right above Theorem 1.2.9, we have the following corollary:

Corollary 1.2.10 *If x_i's are equispaced and ϵ_i's are i.i.d. $N(0, \sigma^2)$, $\hat{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}}_{mG})$ is consistent, provided that $\max(\hat{\boldsymbol{\lambda}}_{mG})/\min(\hat{\boldsymbol{\lambda}}_{mG}) <$ Constant as $n \to \infty$.*

Recall that in Theorem 1.2.6, we established the asymptotic equivalence between $\tilde{\mathbf{f}}_n(\boldsymbol{\lambda})$ and $\hat{\mathbf{f}}_n(\boldsymbol{\lambda})$, under certain conditions. We would like to establish the same property for $\hat{\boldsymbol{\lambda}}_{mG}$. In the next theorem, using the Theorem 1.2.9, we show that under certain conditions, $\tilde{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}}_{mG})$ and $\hat{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}}_{mG})$ are asymptotically indistinguishable.

Theorem 1.2.11 *Under **(A.1)**, **(A.2)**, and **(A.3)**,*

$$\frac{n^{-1}\|\tilde{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}}_{mG}) - \hat{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}}_{mG})\|^2}{L_n(\hat{\boldsymbol{\lambda}}_{mG})} \to 0,$$

where $\hat{\boldsymbol{\lambda}}_{mG}$ is the mGCV choice.

Proof of the above theorem is in Appendix A.6. Together with Theorem 1.2.5, Theorem 1.2.11 demonstrates that if we choose $\boldsymbol{\lambda}$ by minimizing mGCV, the resulting estimate $\hat{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}}_{mG})$ asymptotically minimizes the true loss $L_n(\boldsymbol{\lambda})$, which is a direct measure for estimating $\mathbf{f}_n$ by $\hat{\mathbf{f}}_n(\boldsymbol{\lambda})$.

1.2.7 Asymptotic Optimality of mGCV

In a series of historic papers [30, 31, 19], *asymptotic optimality* has been established for the GCV. In this section, we show the asymptotic optimality of the mGCV in the same sense as in [31]. The asymptotic optimality in adaptive smoothing splines is defined as follows:

$$\frac{L_n(\hat{\boldsymbol{\lambda}}_{mG})}{\inf_{\boldsymbol{\lambda}\in\mathbb{R}_+^{n-1}} L_n(\boldsymbol{\lambda})} \to 1, \quad \text{in probability,} \tag{2.20}$$

where $\hat{\boldsymbol{\lambda}}_{mG}$ is the minimizer of the mGCV function in (2.14).

Under certain conditions, the mGCV method of selecting $\boldsymbol{\lambda}$ is asymptotically optimal as indicated by the following theorem.

Theorem 1.2.12 (Asymptotic Optimality) *Under **(A.1)**, **(A.2)**, and **(A.3)**, $\hat{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}}_{mG})$, where $\hat{\boldsymbol{\lambda}}_{mG}$ is the mGCV choice, is asymptotically optimal as in* (2.20).

Proof of the above theorem is in Appendix A.7.

Theorem 1.2.12 states that under certain conditions, the mGCV choice, $\hat{\boldsymbol{\lambda}}_{mG}$, and the optimal value of $\boldsymbol{\lambda}$ behave the same for sufficiently large n in terms of the corresponding values of the loss. It also says that $L_n(\hat{\boldsymbol{\lambda}}_{mG})$ will tend toward the minimal loss as $n \to \infty$.

Since the mGCV function in (2.14) is *not* convex, minimizing it can be a challenging numerical problem. We suggest to start with a constant vector given by the classical smoothing spline and GCV, then iteratively reduce the value of the mGCV function. We should acknowledge that mGCV may have multiple local minima that may not have the property of the global minima as described in Theorem 1.2.12. Nevertheless, our simulation study in Section 1.2.8 shows effectiveness of the suggested iterative approach. For the suggested iterative approach, we establish the following corollary that parallels Theorem 1.2.12.

Corollary 1.2.13 *Under the previous conditions **(A.1)**, **(A.2)**, and the following condition **(A.3′)**:*

(A.3′) *There exists* $\boldsymbol{\lambda}_n \in \Omega$, *such that* $L_n(\boldsymbol{\lambda}_n) \to 0$, *where* Ω *is defined as* $\Omega = \{\boldsymbol{\lambda} \in \mathbb{R}_+^{n-1} : (\lambda^*, \ldots, \lambda^*)$ *is included*; $mGCV(\boldsymbol{\lambda})$ *as a function of* $\boldsymbol{\lambda}$ *is convex*$\}$, *where* λ^* *is the minimizer of GCV*,

we have,

$$\frac{L_n(\hat{\boldsymbol{\lambda}}_{mG-C})}{\inf_{\boldsymbol{\lambda}\in\Omega} L_n(\boldsymbol{\lambda})} \to 1, \quad \textit{in probability,}$$

where $\hat{\boldsymbol{\lambda}}_{mG-C}$ *is the constrained sequence of minimizer obtained by minimizing the* $mGCV_n(\boldsymbol{\lambda})$ *over* Ω, *where* Ω *is defined in* ***(A.3′)***.

Proof of the above corollary is in Appendix A.8.

Recall that our $\boldsymbol{\lambda}$ has the dimension of the sample size minus one. Although this $\boldsymbol{\lambda}$ allows very flexible adaptation to varying roughness, it may not be computationally efficient. For more efficient computation, one may want to reduce the dimension of $\boldsymbol{\lambda}$ by assuming a step function for $\lambda(x)$ with the number of jumps much less than the sample size. In this case, if we choose the value of the corresponding multivariate smoothing parameter by minimizing the mGCV, the asymptotic optimality still holds, as shown in the following Corollary 1.2.14. Note that Corollary 1.2.14 is for a special case of Theorem 1.2.12.

Corollary 1.2.14 *Suppose that* $\lambda(x) \equiv \lambda_i$ *for* $x \in [s_i, s_{i+1}), i = 1, \ldots, k$, $s_1 = 0$ *and* $s_{k+1} = 1$, $s_i \in \{x_1, \ldots, x_n\}, i = 1, \ldots, k, k \ll n$. *Let* n_i *be the number of design points* $(x's)$ *included in* $[s_i, s_{i+1})$, $i = 1, \ldots, k$. *(Note that* $n_1 + \cdots + n_k = n - 1$.*) Let* $\eta = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ *and* $\boldsymbol{\lambda}_\eta$ *be the multivariate smoothing parameter defined by*

$$\boldsymbol{\lambda}_\eta = (\underbrace{\lambda_1, \ldots, \lambda_1}_{n_1}, \underbrace{\lambda_2, \ldots, \lambda_2}_{n_2}, \ldots, \underbrace{\lambda_k, \ldots, \lambda_k}_{n_k}).$$

Under the conditions ***(A.1)***, ***(A.2)***, *and* ***(A.3)***,

$$\frac{L_n(\hat{\boldsymbol{\lambda}}_{\eta,mG})}{\inf_{\eta\in\mathbb{R}_+^k} L_n(\boldsymbol{\lambda}_\eta)} \to 1, \quad \textit{in probability,}$$

where $\hat{\boldsymbol{\lambda}}_{\eta,mG}$ *is the value of* $\boldsymbol{\lambda}_\eta$ *chosen by minimizing the mGCV function in* (2.14) *with* $\boldsymbol{\lambda}$ *replaced by* $\boldsymbol{\lambda}_\eta$.

The above corollary can be easily proved via the same way as for Theorem 1.2.12. We omit the details.

1.2.8 Finite-Sample Simulations

We have provided theoretical justification on adopting the mGCV. However, considering the fact that mGCV is not convex, minimizing the mGCV can be numerically challenging. In our simulation study, we use the following simple iterative approach:

1. Assume the initial $\boldsymbol{\lambda} = (\lambda, \cdots, \lambda)$, where λ is chosen via the GCV.

2. Define a sequence $\tilde{\boldsymbol{\lambda}}^1(i_0, i_1, \eta) = (\tilde{\lambda}_1^1, \cdots, \tilde{\lambda}_{n-1}^1)$ where we impose the following: if $i_0 \leq i \leq i_1$, $\tilde{\lambda}_i^1 = \lambda_i + \eta$; and $\tilde{\lambda}_i^1 = \lambda_i$, otherwise. Given the $\boldsymbol{\lambda}$ from the step 1, we find i_0, i_1, η, such that the mGCV function

$$n^{-1}\|\mathbf{y}_n - \mathbf{S}_n(\tilde{\boldsymbol{\lambda}}^1)\mathbf{y}_n\|_2^2 / \{n^{-1}\mathrm{tr}[\mathbf{I} - \mathbf{S}_n(\tilde{\boldsymbol{\lambda}}^1)]\}^2$$

   is minimized. This step can be done via an extensive search.

3. Define a sequence $\tilde{\boldsymbol{\lambda}}^2(\eta)$ such that for $1 \leq i \leq n-1$, $\tilde{\lambda}_i^2 = \eta \cdot \lambda_i$ where $\eta > 0$. Given the $\tilde{\boldsymbol{\lambda}}^1$ from the step 2, we compute $\tilde{\boldsymbol{\lambda}}^2(\eta)$ such that the mGCV function

$$n^{-1}\|\mathbf{y}_n - \mathbf{S}_n(\tilde{\boldsymbol{\lambda}}^2)\mathbf{y}_n\|_2^2 / \{n^{-1}\mathrm{tr}[\mathbf{I} - \mathbf{S}_n(\tilde{\boldsymbol{\lambda}}^2)]\}^2$$

   is minimized with respect to η. We declare the convergence and terminate, if η is close enough to 1 or the newly obtained minimum value of mGCV is larger than the one at the previous iteration. Otherwise, bring the $\tilde{\boldsymbol{\lambda}}^2$ back to the step 2. The final $\boldsymbol{\lambda}$ is our estimate.

Of course such a method only guarantees a local minimum. Nevertheless, promising numerical results have been obtained in our simulation study: we found that our method outperforms or performs competitively with other variable penalization methods as well as another popular approach—wavelets. As an alternative, well-known nonlinear optimization algorithms (e.g., Nelder-Mead, quasi-Netown, and conjugate

gradient methods) can be used to minimize the mGCV function. However, researching on the best numerical strategy is beyond the scope of this paper.

Now we compare our method with four competing methods: the weighted smoothing spline (denoted as WSS) proposed by [11], the Loco-Spline (denoted as Loco) proposed by [48], the standard smoothing spline (denoted as SS), and the wavelet shrinkage method. The first two are recently proposed methods for spatial adaptation in the framework of the smoothing spline, and the wavelet shrinkage has emerged to be a powerful nonparametric smoothing method [13]. For the wavelet shrinkage, we choose the Symmlet wavelets with 8 vanishing moments. The coarsest level is set to be $L = 4$—sixteen coefficients associated with scaling functions are kept unchanged. The thresholding method and the estimation of the threshold are identical with the one in [13].

We consider five examples that have been used in literatures on spatial adaptivity: (1) Doppler function [13], (2) Bumps function [13], (3) Heaviside function [38], (4) Mexican hat function [48], and (5) Rapid change function [48]. In each example, we consider 128 and 256 data points sampled regularly on $[0, 1]$. Our results for the first two functions are illustrated in Fig. 2. Each row shows the true function, estimated function, and smoothing parameter (from left to right), together with the results from the standard smoothing spline. We observe that the multivariate smoothing parameter has relatively small values in the region of large local variations, and large values in the region of small local variations. The simulation results are summarized in Table 1. Five methods are compared in terms of the averaged mean squared error (MSE) based on 100 simulations. Overall, we observe that our method outperforms or performs competitively with other competing methods.

Table 1: We compare the proposed method with the weighted smoothing spline (WSS), Loco-spline (Loco), the ordinary smoothing spline (SS), and the wavelet shrinkage in terms of the averaged mean squared error (MSE). All values are based on 100 simulations. The values in the parentheses are standard errors. The minimum in each row is in **bold** face.

Sample size	Signal name	WSS	Loco	SS	Wavelet	Ours
$n = 128$	'Doppler'	3.8794	2.6813	2.9588	0.6443	**0.5380**
		(1.4558)	(0.5371)	(0.3187)	(0.1106)	(0.1275)
	'Bumps'	3.8271	18.4725	27.2375	0.8445	**0.8226**
		(0.8807)	(7.4612)	(1.0001)	(0.1181)	(0.1683)
	'Heaviside'	0.1980	0.5109	0.5217	0.3597	**0.1897**
		(0.0436)	(0.0894)	(0.0813)	(0.0561)	(0.0447)
	'Mexican'	0.1249	**0.0363**	0.0590	0.0954	0.0541
		(0.0481)	(0.0182)	(0.0158)	(0.0124)	(0.0113)
	'Rapid'	0.0021	**0.0007**	0.0008	0.0008	**0.0007**
		(0.0007)	(0.0003)	(0.0003)	(0.0002)	(0.0003)
$n = 256$	'Doppler'	1.3999	1.0370	1.2170	0.5610	**0.4076**
		(0.4667)	(0.4451)	(0.1713)	(0.0865)	(0.0938)
	'Bumps'	2.9107	3.2634	4.4309	0.7391	**0.5561**
		(0.4497)	(0.1967)	(0.4424)	(0.0885)	(0.0894)
	'Heaviside'	0.1174	0.2739	0.3506	0.2894	**0.1158**
		(0.0247)	(0.0629)	(0.0390)	(0.0474)	(0.0227)
	'Mexican'	0.0686	**0.0157**	0.0320	0.0771	0.0264
		(0.0224)	(0.0095)	(0.0079)	(0.0204)	(0.0100)
	'Rapid'	0.0013	0.0004	0.0005	0.0008	**0.0003**
		(0.0004)	(0.0002)	(0.0002)	(0.0001)	(0.0001)

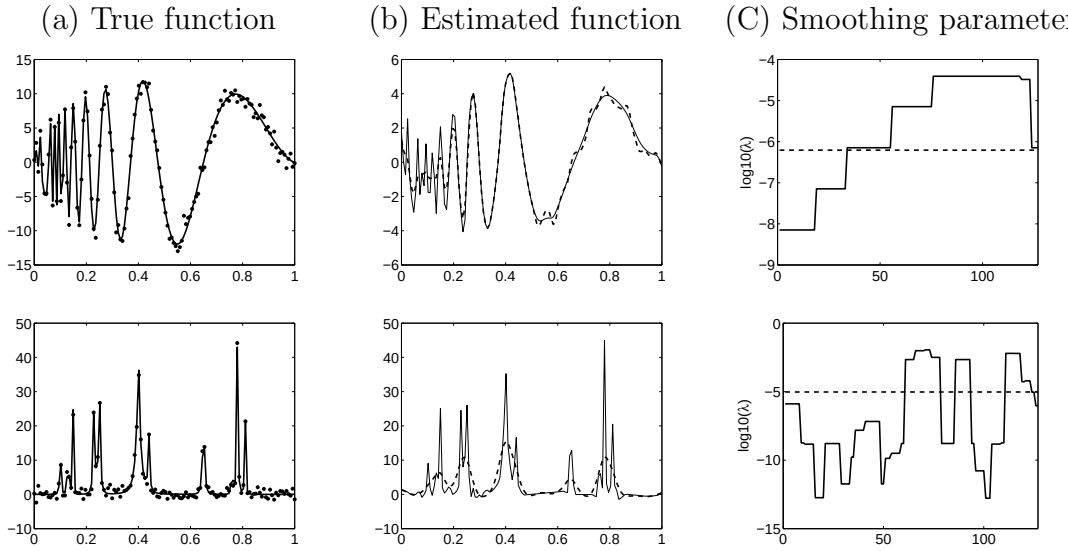


Figure 2: Illustration of the simulation results for the Doppler and Bumps functions (top and bottom rows, respectively). In (a), the true function (solid) is represented, together with noisy observations (dots). In (b), the estimated function using our method (solid) is shown, together with the result from the classic smoothing spline (dashed). We can see that our estimates are much closer to the truth than the traditional smoothing splines estimates. In (c), the corresponding multivariate smoothing parameter (solid) is shown, together with the global smoothing parameter from the standard GCV (dashed).

1.2.9 Conclusion

We develop an adaptive smoothing spline estimator for fitting functions with spatially varying smoothness. The proposed estimator involves a multivariate smoothing parameter $\boldsymbol{\lambda}$. We show that if we choose $\boldsymbol{\lambda}$ via the multivariate version of the generalized cross validation, the resulting estimator is consistent and asymptotically optimal under certain conditions. This provides a theoretical justification of adapting the multivariate version of the generalized cross validation principle in our adaptive penalization framework. We compare our estimator with other competing methods via simulations and show effectiveness of our method.

1.3 Asymptotically Optimal Local Penalty Function for Adaptive Smoothing Splines

1.3.1 Introduction

We consider the following framework of the adaptive smoothing spline:

$$\min_{f \in W_2^n[0,1]} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \int_0^1 \lambda(x) \{f^{(m)}(x)\}^2 dx. \tag{3.21}$$

In this second part of Chapter 1, we derive the local and asymptotically optimal penalty function of $\lambda(x)$ in (3.21). The key for the derivation is to use the connection between the smoothing spline and kernel smoothing. It is well known that smoothing spline is asymptotically equivalent to a variant of kernel smoothing [43, 1]. Using this result, together with the locally optimal bandwidth in kernel smoothing [36], we derive the asymptotically optimal local penalty function. The optimal penalty function is subsequently used for the adaptive smoothing spline.

An algorithm for fitting functions using the proposed method is presented. Since the optimal form of $\lambda(x)$ involves the derivatives of the unknown function, we cannot use it directly. Instead we take an iterative approach: $\lambda(x)$ and $f(x)$ are alternatively updated until convergence is reached.

The rest of this section is organized as follows. In Section 1.3.2, we derive the local and asymptotically optimal penalty function in smoothing spline. In Section 1.3.3, we propose the locally optimal smoothing spline estimator using the optimal penalty function derived in Section 1.3.2. In Section 1.3.4, our method is discussed in terms of the global rate of convergence, and pointwise confidence intervals are constructed. In Section 1.3.5, the proposed method is compared with other competing methods in several simulated examples. We conclude in Section 1.3.6.

1.3.2 Locally Optimal Smoothing Parameter

In this section, we derive the local and asymptotically optimal smoothing parameter in smoothing splines. We focus on the cubic smoothing spline (i.e., with $m = 2$ in (3.21))

which is the most commonly used spline in practice. For the derivation, the key idea is to use the connection between smoothing spline and kernel smoothing. Let $\hat{f}_{\lambda(x)}$ be the cubic smoothing spline estimator with a variable smoothing parameter $\lambda(x)$, defined as the solution to (3.21). The local and asymptotically optimal smoothing parameter with respect to the local MSE (mean squared error), defined as $\mathbb{E}(\hat{f}_{\lambda(x)}(x) - f(x))^2$, is presented in the following theorem:

Theorem 1.3.1 *Assume that $f \in C^4[0,1]$. Let $\hat{f}_{\lambda(x)}$ be the cubic smoothing spline estimator with a variable smoothing parameter $\lambda(x)$, defined as the solution to* (3.21)*. Then the local and asymptotically optimal smoothing parameter (in the sense of asymptotically minimizing local MSE, $\mathbb{E}(\hat{f}_{\lambda(x)}(x) - f(x))^2$) is*

$$\lambda^*(x) = \frac{1}{4}\left(\frac{\sigma^2}{8}\frac{\nu}{\beta_4^2}\right)^{4/9}[f^{(4)}(x)]^{-8/9}n^{-4/9}, \tag{3.22}$$

provided that $f^{(4)}(x) \neq 0$, where $\beta_4 = \frac{1}{24}\int x^4K(x)dx$ and $\nu = \int K^2(x)dx$, where the K is a kernel that satisfies the following moment conditions: $\int_{-1}^{1} K(x)dx = 1$, $\int_{-1}^{1} x^jK(x)dx = 0, j = 1, \cdots, 3$, and $\int_{-1}^{1} x^4K(x)dx \neq 0$. The exact formula of $K(x)$ will be given later in (3.25)*.*

We describe the main idea in the proof.

The proof of Theorem 1.3.1: It is well known (see for example, [52]) that smoothing spline estimator $\hat{f}$ is linear in the observations y_i, and there exists a weight function $W_n(x, x_i)$ such that

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} W_n(x, x_i)y_i. \tag{3.23}$$

The asymptotic behavior of the weight function W_n has been studied by many researchers. In particular, [43] studied the equivalent kernel for standard smoothing splines (i.e., with a global smoothing parameter λ), and showed that the local bandwidth $b(x)$ satisfies $b(x) \sim (\lambda/p(x))^{1/4}$ for the case of cubic smoothing splines (symbol

$\sim$ means "asymptotically equivalent to"), where $p(x)$ is the local density of the design points. This result indicates that for points near x, the bandwidth is determined by the global smoothing factor $\lambda^{1/4}$ as well as the density of the design points near x. Recall that throughout this section, for simplicity, we assume equally spaced design points, i.e., $p(x) \equiv 1$. Under this assumption, Silverman's local bandwidth $b(x)$ becomes a global bandwidth b and it is determined by $\lambda^{1/4}$.

For more general problem, [1] derived the asymptotically equivalent kernels for cubic smoothing splines with a variable smoothing parameter $\lambda(x)$. It is clear that such a solution is also represented as the form of (3.23). In their derivation of the equivalent kernel to W_n, [1] assumed the following conditions:

(i) $\lambda(x)$ is of the form $\lambda(x) = \lambda_0(x)/h^4$, where $\lambda_0(x)$ is a fixed known function independent of n, and h is a parameter that depends on n in such a way that $h \to \infty$ as $n \to \infty$,

(ii) $\lambda_0(\cdot)$ lies in $C^3[0,1]$ and are bounded away from zero, where $C^3[0,1]$ denotes the set of differentiable functions whose third derivative is continuous.

Under the above conditions, the asymptotically equivalent kernel W has been derived as

$$W(x,\cdot) = \frac{h}{2}[\lambda_0(x)\lambda_0(\cdot)]^{-1/8} e^{-h\Phi_0(x,\cdot)} \sin\left(h\Phi_0(x,\cdot) + \frac{\pi}{4}\right),$$

where $\Phi_0(x,\cdot) = \frac{1}{\sqrt{2}} \int_{\min(x,\cdot)}^{\max(x,\cdot)} (1/\lambda_0)^{1/4}$. Thus if x and x_i are close enough, the weight function $W_n(x,x_i)$ can be approximated as

$$W(x,x_i) = \frac{h}{2}[\lambda_0(x)]^{-1/4} e^{-\frac{h}{\sqrt{2}} \cdot \frac{|x-x_i|}{\lambda_0(x)^{1/4}}} \sin\left(\frac{h}{\sqrt{2}} \cdot \frac{|x-x_i|}{\lambda_0(x)^{1/4}} + \frac{\pi}{4}\right). \tag{3.24}$$

In (3.24), if we set

$$b(x) = \lambda_0(x)^{1/4}\frac{\sqrt{2}}{h}, \quad K(u) = \frac{1}{\sqrt{2}} e^{-|u|} \sin(|u| + \frac{\pi}{4}), \tag{3.25}$$

then we have

$$W(x,x_i) = \frac{1}{b(x)} K\left(\frac{x-x_i}{b(x)}\right). \tag{3.26}$$

(3.26) implies that the smoothing spline with a variable smoothing parameter is essentially a kernel estimator with a *variable* bandwidth $b(x)$ and kernel $K(u)$.

On the other hand, an optimal variable bandwidth in kernel smoothing has been shown in [36]. Let $\hat{f}$ be the kernel estimator with a variable bandwidth $b(x)$ defined as

$$\hat{f}(x, b(x)) = \frac{1}{n \cdot b(x)} \sum_{i=1}^{n} K\left(\frac{x - x_i}{b(x)}\right) y_i, \tag{3.27}$$

where K is a kernel function. Denote $C^k[0,1] = \{f : f^{(j)} \text{ is continous, } j = 0, \ldots, k\}$. [36] showed that for $f \in C^k[0,1]$ and $k \geq 0$, if the kernel K satisfies the following moment conditions:

(I) $\int_{-1}^{1} K(x)dx = 1$,

(II) $\int_{-1}^{1} x^j K(x)dx = 0, j = 1, \cdots, k-1$,

(III) $\int_{-1}^{1} x^k K(x)dx \neq 0$,

then the locally optimal bandwidth (in the sense of minimizing $\mathbb{E}(\hat{f}(x, b(x)) - f(x))^2$) is

$$b^*(x) = \left(\frac{1}{2k} \frac{\nu}{\beta_k^2} \frac{\sigma^2}{f^{(k)}(x)^2} \frac{1}{n}\right)^{1/(2k+1)}, \tag{3.28}$$

where $\beta_k = (k!)^{-1} \int x^k K(x)dx$, $\nu = \int K^2(x)dx$, and it is assumed that $f^{(k)}(x) \neq 0$.

Now, using the locally optimal bandwidth $b^*(x)$ in (3.28) and the fact that the smoothing spline with $\lambda(x)$ is asymptotically equivalent to a certain variable kernel smoothing, as shown in (3.26) and (3.27), we derive the local and asymptotically optimal smoothing parameter $\lambda^*(x)$ in the smoothing spline. We found that for K in (3.25), the smallest k that satisfies the three moment conditions (I) $\sim$ (III) is 4. The locally optimal smoothing parameter can be derived as in Theorem 1.3.1 by comparing $b(x)$ in (3.25) and $b^*(x)$ with $k = 4$ in (3.28). □

Remark 1.3.2 *Recall that $\lambda(x)$ is in the form of $\lambda(x) = \lambda_0(x)/h^4$, where $\lambda_0(x)$ is a fixed known function independent of n, and h is a parameter that depends on n in*

such a way that $h \to \infty$ as $n \to \infty$. From Theorem 1.3.1, if $\lambda_0^*(x) \sim [f^{(4)}(x)]^{-8/9}$ and $h^* \sim n^{1/9}$, the local MSE is asymptotically minimized.

In Section 1.3.3, using the Theorem 1.3.1, we propose a locally optimal smoothing spline estimator.

1.3.3 Locally Optimal Smoothing Spline

In this section, using the locally optimal smoothing parameter derived in Section 1.3.2, the locally optimal smoothing spline estimator in turn is derived.

Recall that the cubic smoothing spline estimator with a variable smoothing parameter is defined as the solution to (3.21) with $m = 2$. One challenging problem in using such an estimator is the specification of the penalty function $\lambda(x)$. For this, we derived the local and asymptotically optimal form of $\lambda(x)$ in Theorem 1.3.1. The theorem implies that for the asymptotically optimal estimation of spatially inhomogeneous functions, $\lambda(x)$ should be chosen to vary along with the curvature of the function, in a way that $\lambda(x)$ is proportional to $[f^{(4)}(x)]^{-8/9}$. This leads to assume a specific form of $\lambda(x)$ to be as follows:

$$\lambda(x) = \tau \cdot [f^{(4)}(x)]^{-8/9}, \tag{3.29}$$

where τ is a parameter that depends on n but independent of x. Note that this assumption is different from those of existing methods, in which $\lambda(x)$ has been assumed to be piecewise constant or proportional to $[f^{(2)}(x)]^{-2}$. Using the penalty function $\lambda(x)$ as in (3.29), we propose a new smoothing spline estimator. Here we assume that $f^{(4)}(x) \neq 0$ (Recall Theorem 1.3.1). In fact, if $f^{(4)}(x)$ is zero or close to zero, $\lambda_0(x)$ becomes ill-defined. To avoid this situation in practice, we consider a small constant $\delta > 0$ which is added to the absolute value of $f^{(4)}(x)$. A similar treatment has been considered in [48]. In our numerical experiments in Section 1.3.5, we have observed that it suffices to choose $\delta = 10^{-5} \times \max_{x_i} |f^{(4)}(x_i)|$. Then our $\lambda(x)$ in (3.29) becomes

$$\lambda(x) = \tau \left(\left| f^{(4)}(x) \right| + \delta \right)^{-8/9}. \tag{3.30}$$

Using the new penalty function in (3.30), we propose a new smoothing spline estimator as the minimizer of the following:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \tau \int_0^1 \frac{\{f^{(2)}(x)\}^2}{(|f^{(4)}(x)| + \delta)^{8/9}} dx \tag{3.31}$$

over $C^4[0,1]$. Solving the above problem is challenging. To solve (3.31) directly, one needs to consider the Euler equation [8]. The Euler equation involves the 8th derivative of $f(x)$, which is not convenient to be used here. We suggest the following iterative algorithm to approximate the solution.

1. Let $f_0(x)$ denote the initial estimator of $f(x)$. Take $f_0(x)$ as the cubic smoothing spline estimator, i.e.,

$$f_0(x) = \underset{f \in W_2^2[0,1]}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \int_0^1 \{f^{(2)}(x)\}^2 dx,$$

   where λ is determined via GCV.

2. Let $f_\ell(x)$ denote the estimate of $f(x)$ at ℓth iteration. For $\ell = 1, 2, \cdots$

   (2-1) Given $f_{\ell-1}(x)$, compute $\tilde{\lambda}^\ell(x) = \left(\left|f_{\ell-1}^{(4)}(x)\right| + \delta\right)^{-8/9}$.

   (2-2) Update the estimate of $f(x)$:

$$f_\ell(x) = \underset{f \in W_2^2[0,1]}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \tau \int_0^1 \tilde{\lambda}^\ell(x)\{f^{(2)}(x)\}^2 dx, \tag{3.32}$$

   where τ is determined via GCV.

3. When sequence $\{f_\ell(x)\}$ converges, we conjecture that it converges to the minimizer in (3.31).

In the above iterative algorithm, we have two issues. Firstly, in the step (2-1), we need to estimate $f^{(4)}(x)$. However, it is known that estimating the high order derivative from noisy data is very challenging. Here we take the fourth derivative of the standard smoothing spline estimate which penalizes on the sixth derivative, so

that we have a smooth estimate of the fourth derivative. Generally, one can have a smooth estimate of the mth derivative by penalizing the derivative of order $m+2$ [40]. Such an approach is widely used in estimating the high order derivatives. The implementation is based on the R package "pspline" that is downloadable online.

Secondly, the minimization problem with general $\tilde{\lambda}(x)$ in (3.32) does not have a convenient closed-form solution in the continuum. However, with the approximation that $\tilde{\lambda}(x) = \tilde{\lambda}_i$ for $x \in [x_i, x_{i+1}]$, we can derive the convenient closed-form solution. We utilize the following theorem.

Theorem 1.3.3 *Consider the minimization problem in* (3.21)*, i.e.,*

$$\min_{f \in W_2^2[0,1]} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \int_0^1 \lambda(x) \{f^{(2)}(x)\}^2 dx.$$

Assume that $\lambda(x) = \lambda_i$ *for* $x \in [x_i, x_{i+1}]$ *and let* $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{n-1})$*. The discrete version of the above problem is to find* $f(x_i)$*,* $i = 1, 2, \ldots, n$*, where* $f(x_i)$ *is the value of* $f(x)$ *at* x_i*, and* f *is the minimizer of the above problem. A closed-form solution for an estimator of* $f(x_i)$*,* $i = 1, \ldots, n$ *is as follows:*

$$\hat{\mathbf{f}}_{\boldsymbol{\lambda}} = \frac{1}{n} \left(\frac{1}{n} \mathbf{I} + \frac{h}{3} \mathbf{Q}^T \mathbf{M}^{-1} \mathbf{M}(\boldsymbol{\lambda}) \mathbf{M}^{-1} \mathbf{Q} \right)^{-1} \mathbf{y}, \tag{3.33}$$

where $\hat{\mathbf{f}} = (\hat{f}(x_1), \cdots, \hat{f}(x_n))^T$*,* $\mathbf{y} = (y_1, \cdots, y_n)^T$*,* $\mathbf{I}$ *is the* $n \times n$ *identity matrix,* $\mathbf{M}$ *is the* $(n-2) \times (n-2)$ *matrix with elements* m_{ij} *given by* $m_{ii} = \frac{2h}{3}$ *for* $i = 1, \cdots, n-2$*, where* $h = x_{i+1} - x_i$ *(we assumed the equally spaced design points);* $m_{i,i+1} = m_{i+1,i} = \frac{h}{6}$ *for* $i = 1, \cdots, n-3$*; zeros elsewhere,* $\mathbf{Q}$ *is the* $(n-2) \times n$ *matrix with elements* q_{ij} *given by* $q_{ii} = q_{i,i+2} = \frac{1}{h}$ *and* $q_{i,i+1} = \frac{-2}{h}$ *for* $i = 1, \cdots, n-2$*; zeros elsewhere, and* $\mathbf{M}(\boldsymbol{\lambda})$ *is the* $(n-2) \times (n-2)$ *matrix with elements* m_{ij} *given by* $m_{ii} = \lambda_i + \lambda_{i+1}$ *for* $i = 1, \cdots, n-2$*;* $m_{i,i+1} = m_{i+1,i} = \lambda_{i+1}/2$ *for* $i = 1, \cdots, n-3$*; zeros elsewhere.*

The proof is relegated to Appendix A.9. Using Theorem 1.3.3, with the approximation that $f^{(4)}(x) = f^{(4)}(x_i)$ for $x \in [x_i, x_{i+1}]$, we have the discrete version closed-form solution for (3.32) as in (3.33) with $\lambda_i = \tau \cdot \tilde{\lambda}_i$.

Before we close this section, a fundamental limitation of using of the 4th order derivative needs to be discussed. On one hand, from the derivation of the asymptotic theorem, it appears that it is necessary to use the 4th order derivative. On the other hand, it leads to a restriction on the underlying function $f(x)$ that the proposed method can deal with: the 4th derivative of the underlying function $f(x)$ must be nonzero everywhere. If $f(x)$ has zero 4th derivative within some interval (i.e., it is an up to 3rd degree polynomial within these intervals), the developed asymptotic theory does not apply. So as a precaution, before adopting the proposed method, one should check whether or not a piecewise polynomial up to the 3rd order is sufficient in estimating the underlying function. If the answer is likely yes, then the proposed method is not recommended. Nevertheless, in practice, we can alleviate this limitation by adding a small constant $\delta > 0$ to the absolute value of the 4th derivative as in (3.30).

1.3.4 Further Discussion

In this section, further discussion for our method is presented. In Section 1.3.4.1, our method is discussed in terms of the global rate of convergence. In Section 1.3.4.2, we discuss how pointwise confidence intervals of our method can be constructed.

1.3.4.1 Asymptotics on Global Mean Squared Error

In the previous section, to derive the locally optimal smoothing parameter, the asymptotic formula of the local MSE (i.e., $\mathbb{E}(\hat{f}(x) - f(x))^2$) has been utilized. Another frequently used criterion function is the integrated MSE (i.e., $\int_0^1 \mathbb{E}(\hat{f}(x) - f(x))^2 dx$), which is related to *global* optimal smoothing. For the latter, the optimal rate of convergence for nonparametric regression function is available due to [46]. In this section, our method is discussed in terms of the global rate of convergence.

It is known (e.g., [33]) that in (1.3), if $\min\{\lambda(x)\} = O(n^{-2m/(2m+1)})$ and $\max\{\lambda(x)\} = O(n^{-2m/(2m+1)})$, the asymptotic rate of the integrated MSE is $O(n^{-2m/(2m+1)})$. This

implies that the adaptive smoothing spline with $\lambda(x)$ achieves the *optimal* global rate of convergence [46]. In Theorem 1.3.1, since we assume that $f \in C^4[0,1]$, one may want to use the quintic smoothing spline (i.e., with $m = 4$ in (3.21) and $\lambda(x) =$ constant) rather than the cubic smoothing spline to achieve the optimal global rate of convergence. However, if we use the quintic smoothing spline, its equivalent kernel would involve not only the fourth derivative of f anymore, but also higher order derivative. This is due to the discrepancy of the assumptions for the smoothing splines and kernel smoothing (e.g., the cubic smoothing spline is derived under the assumption of $f \in W_2^2[0,1]$, and its equivalent kernel estimator is of order 4, i.e., f is assumed to be in $C^4[0,1]$). In fact, in the current approach of deriving the equivalent kernel under (3.21), for general m, the underlying f would have $f^{(2m)}$. Hence even we had the equivalent kernel result for a arbitrary m in (3.21), it likely requires the existence of $f^{(2m)}$, hence the optimal global rate of convergence will be $O(n^{-4m/(4m+1)})$ instead of $O(n^{-2m/(2m+1)})$, which is the optimal convergence rate when the optimal penalty with $f^{(m)}$ in (3.21) is used. This shows that if asymptotic global MSE (instead of the local MSE) is adopted, the equivalent kernel approach of this paper will not be able to render a method that has the global optimal rate of convergence. Our method sacrifices some *asymptotic global* rate of convergence. However, we have observed from our simulations that in the finite sample cases, our approach is more advantageous by utilizing the locally optimal penalty function (see Section 1.3.5). In fact, even though the quintic smoothing spline has faster rate of convergence ($O(n^{-8/9})$) than the cubic smoothing spline ($O(n^{-4/5})$), if the sample size is not too large, the finite sample performance of the cubic smoothing spline has been observed to be superior in our simulations. For the various sample sizes, we compared the cubic smoothing spline and quintic smoothing spline using the Doppler function. Table 2 compares the averaged MSE based on 500 simulations. We can see that if $n < 2^{15}$ (roughly $n < 30,000$), the cubic smoothing spline generates smaller MSE. Note many application problems

fall into this case (i.e., $n < 30,000$).

Table 2: For the various sample sizes, the cubic smoothing spline ($m = 2$) and quintic smoothing spline ($m = 4$) are compared using the Doppler function. The averaged MSE based on 500 simulations are shown. The values in the parentheses are standard errors.

n	SS with m=2	SS with m=4
2^7	2.9666 (0.3530)	3.4927 (0.4101)
2^9	0.5812 (0.0420)	0.6621 (0.0543)
2^{11}	0.2162 (0.0123)	0.2331 (0.0128)
2^{13}	0.0720 (0.0037)	0.0739 (0.0036)
2^{15}	0.0226 (0.0011)	0.0217 (0.0010)
2^{16}	0.0126 (0.0005)	0.0116 (0.0005)

1.3.4.2 Pointwise Confidence Intervals

We construct the pointwise confidence intervals for the proposed estimator. This can be done by extending the result in [53], in which pointwise confidence intervals for the standard smoothing spline estimator has been derived using the properties of smoothing splines as Bayes estimates. Denote $CI_{i,\alpha}$ as the pointwise $(1-\alpha)100$ percent confidence intervals at x_i. [53] constructed $CI_{i,\alpha}$ using the GCV smoothing spline estimator $\hat{f}_{\hat{\lambda}}$ as

$$CI_{i,\alpha} = \hat{f}_{\hat{\lambda}}(x_i) \pm z_{\alpha/2}\sqrt{\hat{\sigma}^2 \mathbf{A}_{ii}(\hat{\lambda})}, \tag{3.34}$$

where $\hat{\lambda}$ is chosen by minimizing GCV, $z_{\alpha/2}$ is the $(1-\alpha/2)100$th standard normal percentile, $\hat{\sigma}^2$ is the estimator of σ^2 from $\hat{f}_{\hat{\lambda}}$ given by $\text{RSS}/(n - \text{trace}\boldsymbol{A}(\hat{\lambda}))$ where RSS is the residual sum of squares and $\mathbf{A}$ is the smoothing matrix for the standard smoothing spline, and $\mathbf{A}_{ii}$ is the iith entry of $\mathbf{A}$. [53] showed via simulations that this pointwise confidence intervals also have nice curvewise property: the average coverage probability, $1/n\sum_{i=1}^{n} Pr\{f(x_i) \in CI_{i,\alpha}\}$, is close to $1-\alpha$, which was proved later by [37].

By extending (3.34), we construct the pointwise confidence intervals for the proposed adaptive smoothing spline estimator. Our strategy to extend (3.34) is similar to that in [10]. Let $\boldsymbol{S}(\boldsymbol{\lambda})$ denote the new smoothing matrix defined as

$$\frac{1}{n}\left(\frac{1}{n}\mathbf{I}+\frac{h}{3}\mathbf{Q}^T\mathbf{M}^{-1}\mathbf{M}(\boldsymbol{\lambda})\mathbf{M}^{-1}\mathbf{Q}\right)^{-1}$$

such that $\boldsymbol{S}(\boldsymbol{\lambda})\mathbf{y}=\hat{\mathbf{f}}_{\boldsymbol{\lambda}}$ in (3.33). Then $CI_{i,\alpha}$ can be constructed as

$$CI_{i,\alpha}=\hat{f}_{\hat{\boldsymbol{\lambda}}}(x_i)\pm z_{\alpha/2}\sqrt{\hat{\sigma}^2\boldsymbol{S}_{ii}(\hat{\boldsymbol{\lambda}})}, \tag{3.35}$$

where $\hat{\boldsymbol{\lambda}}$ is the estimator of $\boldsymbol{\lambda}$ from the iterative method suggested in Section 1.3.3 and $\hat{\sigma}^2$ is obtained by $\mathrm{RSS}/(n-\mathrm{trace}\boldsymbol{S}(\hat{\boldsymbol{\lambda}}))$ where RSS is the residual sum of squares from the new adaptive estimate. Fig. 3 shows the estimates (solid) for the Doppler (left) and Bumps (right) functions using the proposed method with 95% confidence intervals (dashed) constructed by (3.35).

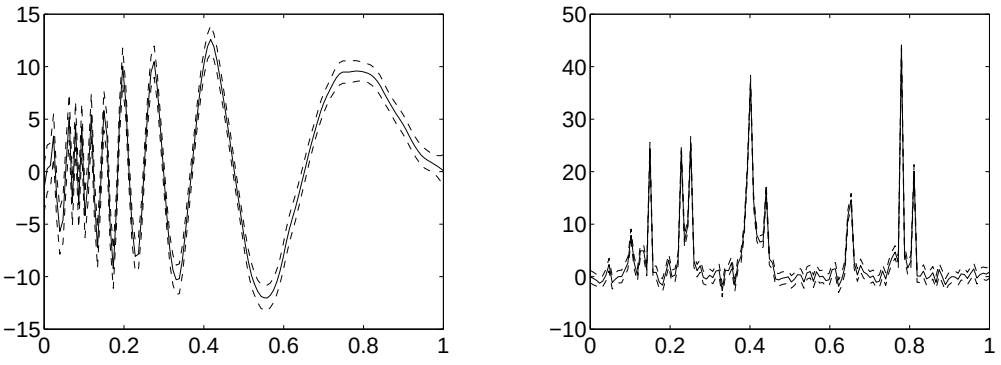


Figure 3: The estimates (solid) for the Doppler (left) and Bumps (right) functions using the proposed method with 95% confidence intervals (dashed).

1.3.5 Simulations

We investigate the performance of the proposed estimator via simulations. Our method is compared with the standard smoothing splines (denoted as SS, we consider the cases of $m=2$ and $m=4$) and two other smoothing spline methods that achieve spatial adaptation: the weighted smoothing splines (denoted as WSS) proposed by [11], and the Loco-Spline (denoted as Loco) proposed by [48]. We choose

these two methods not only for their recency in appearance, but also their representativeness in methodology. The Loco-Spline [48] is one of the most recent published works on variable penalty function in the general smoothing splines framework. [11] takes a very different approach. Instead of a variable smoothing parameter in the penalty term, they consider weights in the least squares term. The weights are then iteratively determined based on a multiresolution analysis of the residuals. These two methods represent the state-of-the-art in spatially adaptive smoothing splines.

We consider the two simulated examples introduced in Section 1.1. The two functions are from the paper by [13] where four spatially inhomogeneous functions were introduced. Considering the recommendation given at the end of Section 1.3.3, we choose the Doppler and Bumps functions to examine here. The functions are defined as follows:

- Doppler function: $f(x) = \sqrt{x(1-x)}\sin(2\pi(1+\epsilon)/(x+\epsilon)), \epsilon = 0.05$.
- Bumps function: $f(x) = \sum h_j K((x-x_j)/w_j), K(x) = (1+|x|)^{-4}$,
  - $(x_j) = (0.1, 0.13, 0.15, 0.23, 0.25, 0.40, 0.44, 0.65, 0.76, 0.78, 0.81)$,
  - $(h_j) = (4, 5, 3, 4, 5, 4.2, 2.1, 4.3, 3.1, 5.1, 4.2)$,
  - $(w_j) = (0.005, 0.005, 0.006, 0.01, 0.01, 0.03, 0.01, 0.01, 0.005, 0.008, 0.005)$.

For each example, we consider $n = 128, 256, 512$ data points sampled regularly on [0,1], and rescale the points so that the signal-to-noise ratio is 7. The noise distributed from $N(0,1)$ is added to the true function. As a measure of the performance, the mean squared error (MSE) is computed. Here MSE is defined as $\text{MSE} = n^{-1}\sum_{i=1}^{n}(f(x_i) - \hat{f}(x_i))^2$. This quantity has been widely used as a performance measure in the literatures, but is sometimes called differently, e.g., TMSE (true mean squared error) in [33]. For each example, we run 100 experiments, and then take the averaged MSE. Table 3 compares the simulation results of SS, WSS, Loco,

and our method. For our method, we use the iterative algorithm proposed in Section 1.3.3. All values for our method in Table 3 are the results after three iterations. It is observed that the convergence is usually achieved within three iterations in nearly all cases.

Table 3: The five smoothing spline methods are compared for the two examples (Doppler and Bumps). The averaged MSE based on 100 simulations are shown. The values in the parentheses are standard errors. For each case, the smallest MSEs are in bold face.

	Setting	Doppler function	Bumps function
$n = 128$	SS with $m = 2$	2.98 (0.32)	27.11 (0.55)
	SS with $m = 4$	3.47 (0.39)	29.29 (0.72)
	WSS	3.66 (1.44)	3.82 (0.88)
	Loco	2.68 (0.53)	18.47 (7.46)
	Our method	**0.95** (0.12)	**3.28** (0.94)
$n = 256$	SS with $m = 2$	1.23 (0.15)	4.35 (0.45)
	SS with $m = 4$	1.46 (0.16)	5.94 (0.31)
	WSS	1.48 (0.49)	2.96 (0.47)
	Loco	1.03 (0.44)	3.26 (0.19)
	Our method	**0.73** (0.09)	**0.99** (0.09)
$n = 512$	SS with $m = 2$	0.58 (0.04)	1.19 (0.14)
	SS with $m = 4$	0.66 (0.05)	1.88 (0.19)
	WSS	0.74(0.23)	1.64 (0.24)
	Loco	0.58 (0.04)	1.14 (0.15)
	Our method	**0.56** (0.04)	**0.97** (0.05)

From Table 3, it is evident that the proposed method outperforms the other competing methods. In fact, our method has the smallest MSE in every case. In particular, our method performs well even when the sample size is relatively small (i.e., $n = 128$ above), while other methods show noticeable inferiority compared to the results of large samples.

The fitting results using our method are illustrated in Fig. 4 and Fig. 5. We consider the worst case, i.e., $n = 128$. The Fig. 4 and 5 show the results for the

Doppler and Bumps functions, respectively. In each figure, in (a), the true function is shown. In (b) and (c), the estimated functions using the standard smoothing spline with $m = 2$ and our method are shown, respectively. In (d), the corresponding spatially adaptive smoothing parameter (i.e., $\lambda(x)$) from our method (solid line) is shown, together with the global smoothing parameter from the standard smoothing spline (dashed line) chosen via GCV. We observe that the spatially adaptive smoothing parameter has relatively small values in the region of large local variations, and large values in the region of small local variations. Moreover, we observe that our estimated smoothing parameter is bumpy. This may not be desirable; smoother penalty function will be more reasonable. This is caused from the difficulty of estimation of the 4th derivative as we already mentioned in Section 1.3.3. Now we suppose that the true 4th derivative is known. Then we have the results as in Fig. 6. In the top row, the estimated Doppler function (in (a)) using the known optimal smoothing parameter (in (b)) are shown. In the bottom row, the estimated Bumps function (in (c)) using the known optimal smoothing parameter (in (d)) are shown. Note that in this figure, the known optimal smoothing parameters in (b) and (d) are rather bumpy. This is not due to granularity (i.e., the finite choice of sample size n), instead, it is a feature of $|f^{(4)}|^{-8/9}$. For very large n, we have observed similar pattern. In these cases, we observe that the estimated functions are almost the same as the true ones. This implies that given a good estimate of the 4th derivative, our method produces nearly the true underlying function.

1.3.6 Conclusion

We develop a new smoothing spline estimator for fitting functions with varying roughness. We first derive the local and asymptotically optimal penalty function $\lambda(x)$, taking advantage of the connection between smoothing spline and kernel smoothing. The derived optimal penalty function is then used for the derivation of the local and

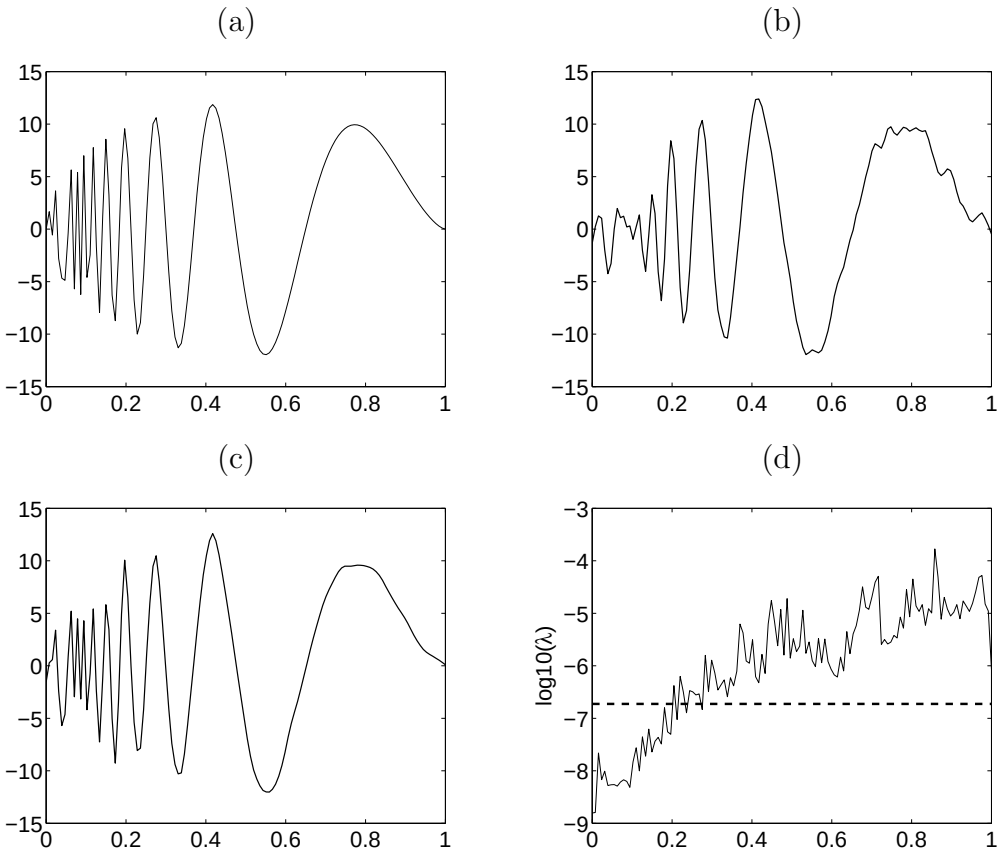


Figure 4: Illustration of the estimation results for the Doppler function. (a): the true function, (b) the estimated function from the standard cubic smoothing spline, (c): the estimated function using our method, (d): the corresponding spatially adaptive smoothing parameter $\lambda(x)$ from our method (solid), together with the global smoothing parameter from the standard smoothing spline chosen via GCV (dashed).

asymptotically optimal smoothing spline estimator. We compare our method with other competing methods via simulations, and show the effectiveness of our method.

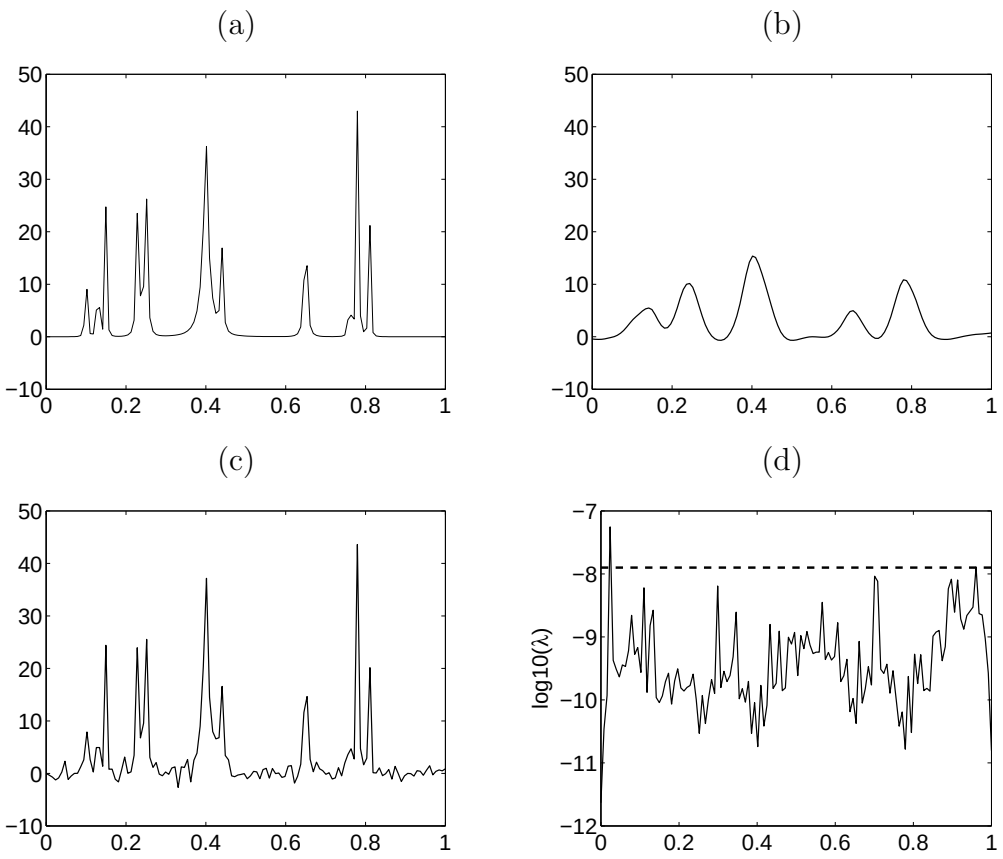


Figure 5: Similar figures as in Fig. 4, see caption of Fig. 4 for description. The underlying noiseless signal is the Bumps function.

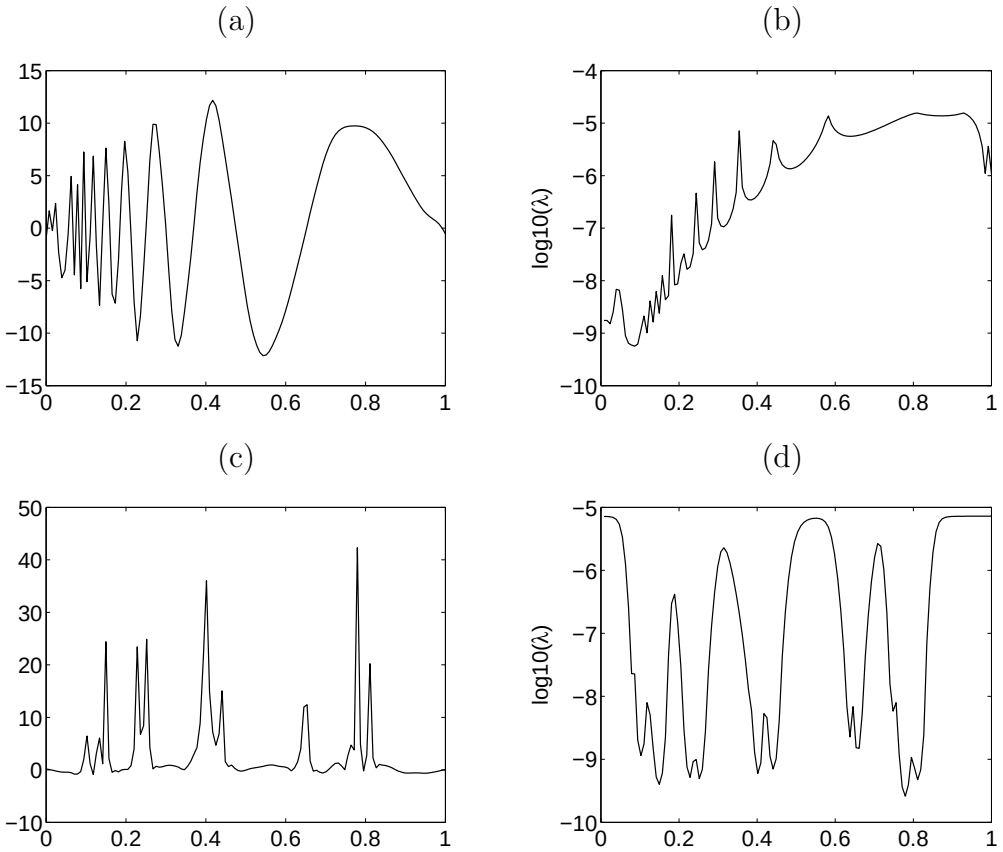


Figure 6: Estimation results using the known optimal penalty function. (a): the estimated function using the known optimal smoothing parameter; i.e., $\lambda(x)$ given in (3.29) and $f(x)$ is the Doppler function given at the beginning of Section 1.3.5, (b): the above mentioned $\lambda(x)$, (c): similar to (a), but with Bumps function, (d): similar to (b), but with Bumps function.

CHAPTER II

A LIPSCHITZ REGULARITY BASED STATISTICAL MODEL, WITH APPLICATIONS IN COORDINATE METROLOGY: CONFIDENCE BANDS, FORM ERROR ESTIMATION, AND OPTIMAL SAMPLING STRATEGY

2.1 Introduction

To ensure the quality of a manufactured part, one needs to decide whether the part meets its design specifications. Dimensional inspection is used to measure the geometric form of a part. By comparing the measured geometry with the design specifications, the form error can be calculated. Acceptance decisions are made based on comparing the error to the specified tolerance. Each part feature, such as straightness, can be measured in a number of different ways. Coordinate Measuring Machines (CMMs) have been widely used in modern manufacturing [12, 32]. According to [6], up to that time, fifteen billion US dollars had been spent for CMMs, with worldwide annual sales being in the range of one billion, and two hundred thousand CMMs being used by a wide range of manufacturers. More recently, annual metrology sales (including CMMs) worldwide were estimated to be around 10 billion US dollars.[1] CMMs typically use a touch probe to collect measurements of the part surface at discrete points. A CMM functions in two basic modes: point-by-point (in which the probe touches the surface once per sampling point) and scanning (in which the probe does not leave the surface). The inspector can determine the number and location of the measurement points. Most CMMs can do point-by-point sampling; only a small

[1] During 2005-2006, Hexagon Metrology believes they have 15% market share (`http://www.hexagon.se/`), with an annual sale of around 1.7 billion US dollars.

proportion of CMMs can implement scanning sampling. There are two intertwined problems:

P1. How do we select the positions of the sample points, so that maximum part information can be obtained from a limited number of points?

P2. Given the limited number of measurement points, how do we construct a confidence band for the examined geometric feature and assess the form error, so that it can be reliably decided whether the part is acceptable?

For P1, sampling strategy for CMM measurements has been extensively studied. Some of the commonly used methods are: uniform, random, and stratified sampling [17]. Three other sampling strategies, Hammersley, Halton-Zaremba, and aligned systematic have been tested at various sample sizes [39]. [29] conclude that using a Hammersley sequence, when compared to a uniform sampling strategy, allows a nearly quadratic reduction in the number of samples needed while maintaining the same level of accuracy. [15] proposed an adaptive sampling strategy which uses an iterative process to select and analyze measurement points. The measurement data are used to develop an interpolating curve which is then used to select subsequent measurement points. The process continues until the measurement converges, with more accurate parts requiring fewer measurement points and less accurate parts requiring more. Unfortunately, programming methods for typical CMMs make this type of iterative measurement very difficult to implement. In the present paper, we do not pursue the idea of adaptive sampling. [23] provides a survey of work done in determining sampling strategies for a variety of two and three dimensional shapes. They conclude that current inspection techniques result in an under-sampling of geometric features on parts with unknown form and measurement errors. They also make a case for the use of intelligent decision systems or procedures for choosing measurement strategies, because best choice is often counter-intuitive. A common limitation of existing studies

is that they do not take advantage of the smoothness property of the surface; our proposed model contributes in this aspect.

For P2, there have been extensive studies for form error assessment using CMM measurements. Two most popular methods are the minimum zone (MZ) method and the orthogonal least squares (OLS) method [14, 41]. The MZ method finds the maximum inscribing and minimum circumscribing features that bound all the CMM data and uses the orthogonal width to estimate the form error. The OLS method fits an ideal feature to CMM data by minimizing the sum of squared orthogonal residuals and uses the range of the resulting orthogonal residuals to estimate the form error. Recently, [56] proposed the Gaussian process model, in which a sequence of CMM measurements are decomposed into three components: global trend, spatially correlated systematic errors, and spatially uncorrelated random errors. Based on the Gaussian process models of both systematic and random errors, a part surface is predicted, and the form error is subsequently estimated by finding the maximum inscribing and minimum circumscribing geometry that bounds all points on the predicted surface.

We propose a wavelet-based method to deal with the two issues above. Through the proposed method, one can construct confidence bands, assess form error, and determine the sampling positions. Differing from the existing literature, we use data to motivate our model, instead of imposing a model to the data. It turns out that a particular model that is based on the wavelet transform is an ideal model for CMM measurements. We start with studying the properties of the CMM measurements. It is found that the Lipschitz regularity holds for the CMM data, when the scanning method is used. It is also known that some specially designed wavelets are Lipschitz, and the wavelet coefficients of a Lipschitz curve decay exponentially as a function of the scale index. Combining these two, we introduce a statistical model for the CMM data. Based on this statistical model, one can construct a confidence band.

Furthermore, form error assessment can be carried out. The proposed model also provides a way to determine the sampling positions. It turns out that for this model, the optimal sampling positions should be the maximum point of the scaling functions in the corresponding wavelet transform. We use both real and synthetic data to test the proposed model. The Lipschitz regularity of the measured surface is justified by considering the surface properties resulting from typical machining operations.

We describe a hypothetical scenario, to demonstrate how to adopt our model in engineering application. Suppose there are many surfaces to be examined for flatness (via straightness). We confirm the Lipschitz property of surfaces made by the manufacturing process (see Section 2.2.1), by taking dense data on a sample part, and studying the smoothness property of these surfaces via the wavelet transform method that will be described later. When the Lipschitz property is confirmed, one can adopt our model to construct a confidence band, assess the form error, and most importantly, determine the optimal positions where a small number of samples need to be taken. By reducing the sampling positions, the entire inspection process is expedited. The smoothness property of the surface (across multiple parts) will not change unless the manufacturing process is changed (see Section 2.2.1).

The rest of this chapter is organized as follows. In Section 2.2, the formulation of the CMM data is presented; the measurements are statistically modeled via wavelet decomposition. Based on such a model, in Section 2.3, we propose a wavelet-based method to construct a confidence band for the measurements, to assess form error, and to determine the optimal sampling positions. Our method is validated with real and synthetic data in Section 2.4 and Section 2.5, respectively. In Section 2.6, some justification and future extension of our method are discussed. Finally, we conclude in Section 2.7.

2.2 Background and Formulation

In this chapter, we focus on a class of problems that arises in CMM measurements. As an example, Fig. 7 contains two subfigures, which are real measurements taken in a CMM facility that is located at Sandia National Laboratories in New Mexico. The data are taken on the same surface, along a straight line, with an objective to

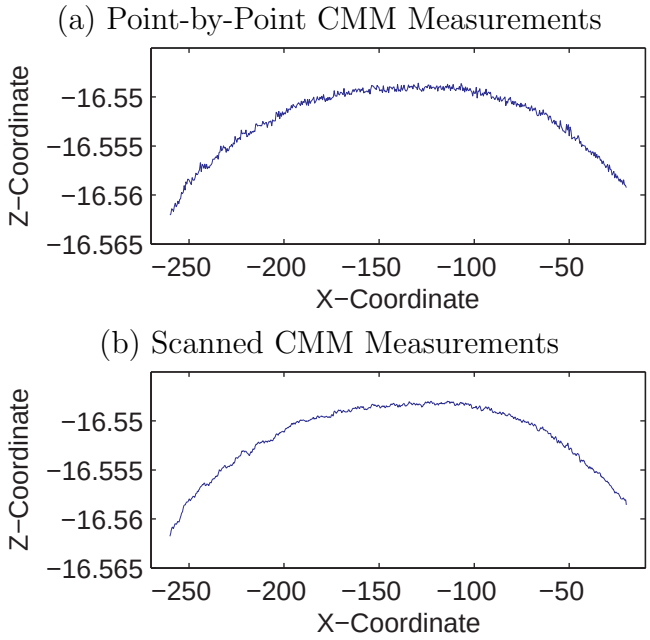


Figure 7: Two real sequences of CMM measurements. The lengths are 512 and 1201, respectively. Only the x and z-coordinates are of interest (because we consider the straightness of the line). The unit for both axes is 10^{-3} meter.

measure the straightness. They are obtained in two approaches: a point-by-point scheme and a scanning scheme. In both subfigures, the x-axis is along the surface, while the vertical axis is normal to the surface.

In this chapter, we first consider the mathematical property of the surface, on which the CMM data are taken. It is observed that if one takes a straight line on the surface, the resulting curve is *uniformly Lipschitz*. Because of such a property, the wavelet orthonormal bases can be utilized to create a model for the CMM data. The statistical model imposes statistical distributions on the wavelet coefficients. In our approach, the adoption of wavelets is a consequence, instead of a preassumption. We

describe details in the following subsections.

The remaining of this section is organized as follows. Lipschitz property is reviewed in Section 2.2.1; we also discuss justification of such an assumption on part surfaces. The wavelets basis functions and the properties of wavelet coefficients for Lipschitz functions are presented in Section 2.2.2 and 2.2.3, respectively. A statistical model for CMM measurements (which is taken from a Lipschitz function) is presented in Section 2.2.4. The Sections 2.2.1 through 2.2.3 are based on continuum. Due to the nature of CMM measurements, Section 2.2.4 is written for discrete data.

2.2.1 Lipschitz Regularity and Justification Related to Part Surface

Recall a function $f(\cdot)$ is *pointwise Lipschitz* γ ($\gamma > 0$) [34] at point x if there exists a constant $K > 0$ and a polynomial $p_x(t)$ in the neighborhood of x, of degree $\lfloor \gamma \rfloor$ (i.e., the largest integer no larger than γ) such that

$$\forall\, t \in \mathbb{R}, \qquad |f(t) - p_x(t)| \leq K|t - x|^{\gamma}. \tag{2.36}$$

A function is *uniformly Lipschitz* γ over $[a, b]$ ($a \leq b$) if for all $x \in [a, b]$, there is a constant K (that is independent of x) such that (2.36) holds. The *Lipschitz regularity* of a function f is the supreme of γ such that f is uniformly Lipschitz γ. Lipschitz regularity can measure the smoothness of f. The essence of Lipschitz regularity is how f can be locally approximated by a polynomial function—whose degree is a natural indicator of the smoothness.

The CMM is often utilized to measure the straightness of a machined surface. The act of machining intuitively leads to a surface that is locally polynomial. Below, we discuss physical justification on this assumption. A test on real data will be presented in Section 2.4. Manufacturing errors in machining are attributed to geometrical errors in the machine (such as quasi-static errors due to machine error motions, thermally induced geometrical errors) [44]. In addition, cutter/material interactions will also produce surface roughness and finish imperfections [5]. Surface roughness and finish

in machining operations, such as milling, are typically on the order of 0.8 to 6.3 micrometer roughness average [55]. Another contributor to form error is tool/cutter deflections. This "copying" error [51] can be modeled as a linear spring-mass-damper system [58], and can be minimized with proper selection of machining parameters [42]. Typical tolerances specified on machined parts are much larger than the surface roughness and finish [22], for example, on the order of 100 micrometer for geometry fabricated by end-milling. In typical machining practice and measurement equipment practice, the machine geometrical errors are mapped at discrete points, with linear interpolation between the mapped points [57]. This makes the machine geometry follow a piecewise linear path, which is also piecewise polynomial.

2.2.2 Orthonormal Wavelets Basis

We now review some basics of wavelets. For $f(x)$, its wavelet decomposition has a form:

$$f(x) = \sum_{j \in \mathcal{I}_L} \alpha_j \phi_j(x) + \sum_{i>L} \sum_{j \in \mathcal{I}_i} \beta_{ij} \psi_{ij}(x), \tag{2.37}$$

where $\phi_j(x)$ are scaling functions at the coarsest scale, $\psi_{ij}(x)$ are wavelet functions, L is the coarsest scale, $\mathcal{I}_L$ is the set of location indices at the coarsest scale while $\mathcal{I}_i$ is the set of location indices at scale i, finally, i and j are the scale and location indices, respectively. Note that $\phi_j(x)$ and $\psi_{ij}(x)$ can be derived by shifting and scaling: $\phi_j(x) = \phi(x - j \cdot c)$ and $\psi_{ij}(x) = \psi(2^{i-L} \cdot x - j \cdot c) \cdot 2^{\frac{1}{2}(i-L)}$. Mostly, we choose ϕ and ψ with finite support; e.g., the Daubechies' wavelets. It is known that if ψ has p vanishing moments, then ϕ and ψ are roughly Lipschitz γ with $\gamma \approx 0.2p$. We refer to Chapter 7 of [34] for more details. We will use the Lipschitz condition of Daubechies' wavelets. Moreover, we will need the following result.

Theorem 2.2.1 *If ψ is uniformly Lipschitz γ with constant K, then $\psi_{ij}(i > L)$ is also Lipschitz γ with constant $K \cdot 2^{(i-L)(\gamma+\frac{1}{2})}$.*

Proof. Recall we have

$$|\psi(t) - p_x(t)| \leq K|t-x|^\gamma,$$

where $p_x(t)$ is a polynomial given in (2.36). One can easily verify the following:

$$|2^{\frac{1}{2}(i-L)}\psi(2^{i-L}\cdot t) - p_{x\cdot 2^{(i-L)}}(2^{i-L}\cdot t)\cdot 2^{\frac{1}{2}(i-L)}| \leq K\cdot 2^{(i-L)(\gamma+\frac{1}{2})}|t-x|^\gamma.$$

The above is equivalent to the fact that ψ_{ij} is Lipschitz γ with $2^{(i-L)(\gamma+\frac{1}{2})}K$. □

2.2.3 Properties of Wavelets Coefficients

We review an important property of wavelet coefficients. This property is the foundation of our statistical model. If function f is smooth, then the wavelet coefficients β_{ij} decay exponentially as a function of the scale: For example, *if f is uniformly Lipschitz γ over $[0,1]$, then we have $|\beta_{ij}| < A\cdot 2^{-\sigma i}$, where A is a constant and $\sigma = \gamma + 1/2$* [34, Section 6.1, Theorem 6.3]. We utilize the above property to construct a system, which can generate Lipschitz γ curves. Note that (2.37) involves infinite scales. In practice, we do not need to consider a function with infinite fine scales—we sacrifice some mathematical rigor here. We consider a truncated case: for $L' > L$, we consider

$$f(x) = \sum_{j\in I_L}\alpha_j\phi_j(x) + \sum_{i=L+1}^{L'}\sum_{j\in I_i}\beta_{ij}\psi_{ij}(x), \tag{2.38}$$

where L' determines how well (i.e., up to which fine scale in the multiresolution analysis) the experimenter wants our model to approximate the true curve. Note that for fixed $i \in \{L+1, L+2, \ldots, L'\}$, and fixed x, $\sum_{j\in I_i}\psi_{ij}(x)$ only has finite number of nonzero terms. In particular, if we consider Daubechies' wavelets with p vanishing moments, the number of nonzero terms is $2p+1$. We will need the following property.

Theorem 2.2.2 *In* (2.38)*, if we impose $|\beta_{ij}| \leq A\cdot 2^{-(\gamma+\frac{1}{2})(i-L)}$, and ψ, ϕ are Lipschitz γ with constant K, then $f(x)$ is Lipschitz γ with constant that is determined by $K, A, L'-L$, and α_j's.*

Proof. Similar to the proof for Theorem 2.2.1, we can show that:

(1) $\sum_{j\in I_L} \alpha_j \phi_j(x)$ is Lipschitz γ with constant $K \cdot \sup_k \sum_{j=k}^{k+2p} |\alpha_j|$.

(2) For fixed $i \in [L{+}1, L']$, $\sum_{j\in I_i} \beta_{ij}\psi_{ij}(x)$ is Lipschitz γ with constant $(2p{+}1)\cdot K \cdot A$. (Note that we need to call Theorem 2.2.1 to establish this result.)

Overall, $f(x)$ is Lipschitz γ with constant: $K\cdot\sup_k \sum_{j=k}^{k+2p} |\alpha_j| + (2p{+}1)\cdot K\cdot A\cdot(L'-L)$.

□

2.2.4 Proposed Statistical Model

Taking advantage of Theorem 2.2.2, we establish the following statistical model for the CMM measurements. Recall that (2.38) contains a model of $f(x)$ in continuum. CMM measurements are always discrete. Section 2 of [56] gives a nice description on CMM data modeling; this paper adopts a similar approach. The sampling points are denoted by $S_\ell, \ell = 1, 2, \ldots, N$, where N is the sample size. Let Y_ℓ denote the relevant CMM measurement at S_ℓ, we assume that

$$Y_\ell = f(S_\ell) + \varepsilon_\ell, \quad 1 \le \ell \le N, \tag{2.39}$$

where $f(\cdot)$ is given in (2.38) and ε_ℓ's are measurement errors. If in relative to $f(S_\ell)$, error ε_ℓ is negligibly small, the property of Y_ℓ's is mainly up to the underlying function $f(x)$. Our model is intended for such a situation.

Now we focus on the situation when ε_ℓ's are negligibly small and $f(x)$ is Lipschitz. We adopt Daubechies' wavelets with p vanishing moments. Other wavelets may be chosen, as long as they satisfy the Lipschitz condition and the finite-support condition that we required. Moreover, we impose that

$$|\beta_{ij}| \le A \cdot 2^{-(\gamma+\frac{1}{2})(i-L)}, \tag{2.40}$$

where A is a prescribed constant, and γ is a prescribed regularity index. The CMM measurements Y_ℓ's are given in (2.39) with $f(x)$ specified in (2.38); i.e., we have

$$Y_\ell = f(S_\ell) + \varepsilon_\ell = \sum_{j \in I_L} \alpha_j \phi_j(S_\ell) + \sum_{i=L+1}^{L'} \sum_{j \in I_i} \beta_{ij} \psi_{ij}(S_\ell) + \varepsilon_\ell, \quad \ell = 1, 2, \ldots, N. \quad (2.41)$$

There is no particular restriction on α_j; hence the model can accommodate functions with various shapes.

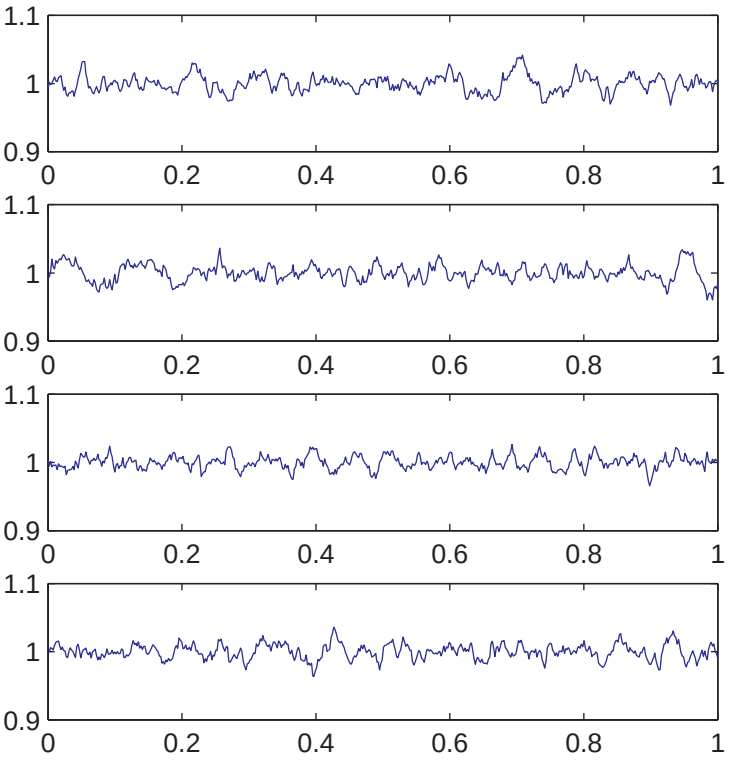


Figure 8: Four simulated Lipschitz-0.5 curves by utilizing Daubechies' symmlets.

For illustration purpose, Fig. 8 presents four curves by setting $\beta_{ij} \sim \frac{1}{10} 2^{-\sigma(i-L)} \cdot N(0,1)$, where the total length of signal is $N = 2^9 = 512$ and the coarsest scale is $L = 4$. The Daubechies' nearly symmetric wavelets (*a.k.a.* Symmlets) with six vanishing moments is adopted. We choose $\sigma = 1$ (correspondingly, $\gamma = 0.5$). Coefficients of all scaling functions (α_j's) are set to be zero—This is why these simulated curves look flat, while the measured data in Fig. 7 are not. Despite the shape (i.e., focusing on the smoothness), we find that such a curve resembles the real data that are obtained via CMMs.

The aforementioned β_{ij}'s satisfy the normal distribution, which is not strictly bounded. However, it is known that normal distribution is highly concentrated around zero. For example, it is extremely unlike to have a realization of a normally distributed random variable that is six standard deviation away from its mean. Hence in simulation, one can treat it as a bounded random variable without much loss. The QQ plots that will be displayed in Section 2.4 show that for real CMMs data, when the measurement errors are negligibly small (e.g., scanning data), the β_{ij}'s do satisfy the normal distribution.

2.3 Proposed Methods

We describe our strategy for three key problems in analyzing CMM data (with regard to straightness). Our construction of confidence bands is described in Section 2.3.1. When reduced sample size is pursued, an optimal sampling strategy is derived in Section 2.3.2. Form error assessment is studied in Section 2.3.3.

2.3.1 Confidence Band

To construct a confidence band, we first specify the baseline (denoted by $b(x)$) of the band. When the half-width (denoted by w) is given, the confidence band is simply $b(x) \pm w : \{(x, y) : b(x) - w \leq y \leq b(x) + w\}$. We first describe our methods of specifying $b(x)$ in Section 2.3.1.1. The w is determined in Section 2.3.1.2. How to use the constructed confidence band towards the part acceptance is discussed in Section 2.3.1.3. Our purpose is to decide the straightness, hence the assumption on constant bandwidth (w) is justifiable.

2.3.1.1 Baseline

Recall the CMM model in (2.41). Nearly all wavelet transforms are based on equally spaced samples. Without loss of generality, here we assume that $S_\ell = \ell/N, \ell = 1, 2, \ldots, N$. (Note when S_ℓ's are not equally spaced, as long as N is large and the

sampling is dense everywhere, one may adopt an interpolation strategy to transform them into equally spaced samples.) Estimating the baseline is essentially a smoothing operation. So we assume standard properties on ε_ℓ's, e.g., they are white noise. Note that we are interested in the case when ε_ℓ's are very small in relative to the first two terms in (2.41). The deviation of ε_ℓ's from the above assumption will not be devastating. We describe two approaches to specify $b(x)$.

- We can simply use the measurement sequence as the baseline. The disadvantage is that the measured curve can be noisy. The justification of this approach is that for each position, the observed is a point estimate of $b(x)$ at the corresponding location.

- We first carry out a wavelet transform of the measured data. Suppose that the wavelet coefficients are $\{\hat{\alpha}_j, \hat{\beta}_{ij}, i = L+1, \cdots, L'\}$. The hat indicates that they are computed from the observations. We then set $\hat{\beta}_{ij} = 0$ for $i \geq L_0$, where $L + 1 \leq L_0 < L'$. Such an approach is identical with the wavelet shrinkage [13] method, which has many nice statistical properties. We then apply *inverse wavelet transform* to the shrunken coefficients. The result is our baseline estimate $b(x)$.

In our numerical study, we adopt the latter approach, because it renders a smoother baseline.

2.3.1.2 Width of the Confidence Band

To determine w, there are at least two approaches. The first one is conservative; it furnishes a wider band. The second one is more accurate, given that the assumed statistical model is close to the reality.

- Recall that for any x, ignoring the small error term, we have

$$f(x) = \sum_{j \in \mathcal{I}_L} \alpha_j \phi_j(x) + \sum_{i > L} \sum_{j \in \mathcal{I}_i} \beta_{ij} \psi_{ij}(x).$$

We treat $\sum_{j\in\mathcal{I}_L}\alpha_j\phi_j(x)$ as the baseline; because this part is not random. The random component (denoted by $u(x)$) becomes

$$u(x) = f(x) - \sum_{j\in\mathcal{I}_L}\alpha_j\phi_j(x).$$

By design, we have $|\beta_{ij}| \le A\cdot 2^{-\sigma i}$, which leads to

$$|u(x)| \le \sum_{i>L}\sum_{j\in\mathcal{I}_i}|\psi_{ij}(x)|\cdot A\cdot 2^{-\sigma i}.$$

Note that $\psi_{ij}(x)$ is a wavelet function, which is a result of scaling and shifting of a standard (finite supported) function ψ: $\psi_{ij}(x) = 2^{i/2}\psi(2^i x - j\cdot c)$. For fixed i, the following should be upper bounded:

$$2^{-i/2}\sum_{j\in\mathcal{I}_i}|\psi_{ij}(x)| = \sum_{j\in\mathcal{I}_i}|\psi(2^i x - j)| < \text{ constant} = C_1.$$

Hence as long as $\sigma > 1/2$, we have that

$$|u(x)| \le C_1\sum_{i>L}2^{i/2}\cdot A\cdot 2^{-\sigma i} = C_1\cdot A\cdot\sum_{i>L}2^{-i(\sigma-1/2)} < \text{ constant}.$$

The above also points out a way to compute the upper bound.

- The aforementioned gives the worst-case bound. We can derive a more accurate bound via simulation. Recall that

$$u(x) = \sum_{i>L}\sum_{j\in\mathcal{I}_i}\beta_{ij}\psi_{ij}(x).$$

For fixed x, $\psi_{ij}(x)$'s are fixed. We choose $\beta_{ij}\sim A\cdot 2^{-\sigma(i-L)}\cdot N(0,1)$, where L is the coarsest scale and $i > L$, A is fixed. Let $\|u(x)\|_\infty$ denote the supreme of $|u(x)|$: $\|u(x)\|_\infty = \sup_x|u(x)|$. Note $\|u(x)\|_\infty$ is a random number. Its percentiles (e.g., the 99th percentile) can be estimated via simulation.

In simulation study, it occurs that the bounds in the former method is usually 2 or more orders of magnitude larger than the counterpart that is given by the latter

method. This indicates that the former method is *too* conservative, and contains too many zero-probability events. Furthermore, in our simulation study with real data sets, in almost all the time we observed that the β_{ij}'s at fixed i behave like normally distributed random variables, referring to Section 2.4 and Fig. 13. Hence in our simulation, we choose the latter.

2.3.1.3 Use of Confidence Bands

Once a confidence band is constructed, one can make part acceptance decision accordingly. Acceptance of parts is based on an agreement between the producer and the consumer. The consumer has a functional need for the nominal measured quantity; where if the part deviates too far from the nominal value, the part will not function. With *simple acceptance* [4], if the measurement falls outside the tolerance zone, the part is rejected. Frequently, the customer and producer will agree that the measurement equipment uncertainty must be some ratio smaller than the tolerance zone, in order to use *simple acceptance*. Another frequently used standard [3] incorporates a decision rule based on the probability of false acceptance (probability that a part measured as *acceptable* is actually non-conforming is $< 2\%$). The derived confidence band is compared with the tolerance specification in order to determine the acceptance.

The tolerance is also often specified by consumers as *form error*. Given a confidence band, one can estimate the form error using, e.g., the minimum zone approach. The result is compared with the tolerance level that is given by the consumer and a decision can be made accordingly. A potential limitation of this approach is discussed in Section 2.3.3.

2.3.2 Optimal Sampling Positions

We now consider how to determine the optimal sampling positions for a new surface that needs to be examined. We will address this problem in two steps. Firstly, we

suppose that a few samples have been taken, and we need to estimate the baseline. We introduce our interpolating scheme, which adopts the minimum energy principle. A closed-form solution is presented (Section 2.3.2.2). Secondly, based on the result from the first step, we consider what will be the optimal sampling positions. Under our framework, we argue that the optimal sampling positions should be chosen at the positions where the scaling functions take the maxima (Section 2.3.2.3).

2.3.2.1 Problem Description and Notations

Recall in Section 2.2.4 that the CMM measurements are modeled as $Y_\ell = f(S_\ell), 1 \leq \ell \leq N$, where S_ℓ is a sampling position on which a discrete wavelet transform is based (We assume that $\varepsilon_\ell \approx 0$.). From this point, we let $\{S_\ell\}_{\ell=1}^N$ denote a dense enough set of measurements (i.e., N is large enough), while a subset of $\{S_\ell\}_{\ell=1}^N$ (i.e., much smaller number of samples) is denoted by $\{s_\ell\}_{\ell=1}^n$ where $n \ll N$; recalling that we assigned $S_\ell = \frac{\ell}{N}$ in Section 2.3.1.1, so $\{s_\ell\}$ is a subset of $\{\frac{1}{N}, \frac{2}{N}, \ldots, 1\}$. The objective is to find the optimal positions of $\{s_\ell\}$. According to (2.41), we have a system of linear equations for the measurements at $\{s_\ell\}$:

$$y_\ell = f(s_\ell) = \sum_{j \in \mathcal{I}_L} \alpha_j \phi_j(s_\ell) + \sum_{i>L}^{L'} \sum_{j \in \mathcal{I}_i} \beta_{ij} \psi_{ij}(s_\ell), \qquad \ell = 1, 2, \ldots, n. \tag{3.42}$$

Note the error ε_ℓ is tentatively left out; because we focus on the true surface at this moment. If we consider the dense set of measurements at $\{S_\ell\}$, we have a *complete* system of equations:

$$Y_\ell = f(\ell/N) = \sum_{j \in \mathcal{I}_L} \alpha_j \phi_j(\ell/N) + \sum_{i>L}^{L'} \sum_{j \in \mathcal{I}_i} \beta_{ij} \psi_{ij}(\ell/N), \qquad \ell = 1, 2, \ldots, N. \tag{3.43}$$

Note that the equations in (3.42) is a subset of equations in (3.43). We introduce notations that will facilitate future discussion. Let $Y = (f(1/N), f(2/N), \ldots, f(1))^T$, $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{2^L})^T$, $\beta = (\beta_{ij})^T$—i.e., β is a column vector that contains all

β_{ij}'s. Let $\mathbf{\Phi}_1 = (\phi_j(\ell/N))_{\ell j}$ and $\mathbf{\Phi}_2 = (\psi_{ij}(\ell/N))_{\ell,ij}$—matrix $\mathbf{\Phi}_2$ contains all values $\psi_{ij}(\ell/N)$ in (3.43). The system in (3.43) can be rewritten as

$$Y = \mathbf{\Phi}_1\alpha + \mathbf{\Phi}_2\beta. \tag{3.44}$$

Let $y = (y_1, y_2, \ldots, y_n)^T$; i.e., y is a subset of Y, consisting of the measurements at $\{s_\ell\}_{\ell=1}^n$. We use y^c to denote the complement of y within Y. Let Φ_1 and Φ_2 denote the subset of rows of $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$, whose membership is consistent with $\{s_1, s_2, \ldots, s_n\}$ being a subset of $\{1/N, 2/N, \ldots, 1\}$. The equations (3.42) is equivalent to the following:

$$y = \Phi_1\alpha + \Phi_2\beta. \tag{3.45}$$

Moreover, let Φ_1^c and Φ_2^c denote the matrices made by the remaining rows of $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$ however not included in Φ_1 and Φ_2, respectively. The complete system in (3.44) (or (3.43)) can be written as follows:

$$\begin{pmatrix} y \\ y^c \end{pmatrix} = \begin{pmatrix} \Phi_1 & \Phi_2 \\ \Phi_1^c & \Phi_2^c \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}. \tag{3.46}$$

2.3.2.2 Wavelet-Based Random Curve Interpolating Algorithm

In this section, we propose an interpolating algorithm which can be used to estimate the baseline when only a small number of measurements are available (i.e., when only the measurements at $\{s_\ell\}_{\ell=1}^n$, not at $\{S_\ell\}_{\ell=1}^N$, are available). In (3.45), suppose y and β are known. The only unknown variable is α, which satisfies $\Phi_1\alpha = y - \Phi_2\beta$. The minimum ℓ^2 norm solution for α corresponds to the following optimization problem:

$$\begin{aligned} \min_{\alpha} \quad & \|\alpha\|_2^2 \\ \text{subject to} \quad & \Phi_1\alpha = y - \Phi_2\beta. \end{aligned} \tag{3.47}$$

We have the following lemma for the solution to the above problem.

Lemma 2.3.1 (3.47) *is a quadratic programming problem, which has the closed-form solution:*

$$\alpha = \Phi_1^T(\Phi_1\Phi_1^T)^{-1}(y - \Phi_2\beta). \tag{3.48}$$

The proof has been relegated to Appendix A.10.

Using Lemma 2.3.1, we propose an interpolating algorithm as follows (Note y is available):

1. Generate β such that each β_{ij} satisfies (2.40). This ensures the Lipschitz property. (In order to generate β_{ij}, parameters A and σ need to be specified. Recall that we first take dense measurements (i.e., $\{S_\ell\}_{\ell=1}^N$) from a few parts to ensure Lipschitz property. The parameters A and σ can be estimated from the measurements at $\{S_\ell\}_{\ell=1}^N$. Estimation of A and σ from such measurements will be described in Section 2.4.2.)

2. Apply Lemma 2.3.1 to obtain α. The interpolated function at the dense set of sampling positions (i.e., $\ell/N, \ell = 1, 2, \ldots, N$) is obtained by (3.44).

The above method will be called *minimum energy interpolation*. Note when $n \ll N$, one should let the aforementioned algorithm substitute the method in Section 2.3.1.1. By letting $N \to \infty$, one interpolates $f(x)$ nearly everywhere.

We justify the adoption of minimum energy interpolation through the following two arguments.

- We first use illustration to establish an intuition. To do so, we find an α that satisfies (3.45), however it is not required to be a solution in (3.47). Fig. 9 plots a minimum energy interpolating curve in black, together with two other non-minimum-energy random interpolating curves (in red and green). It is observed that minimum energy interpolating curves are closer to observations in metrology.

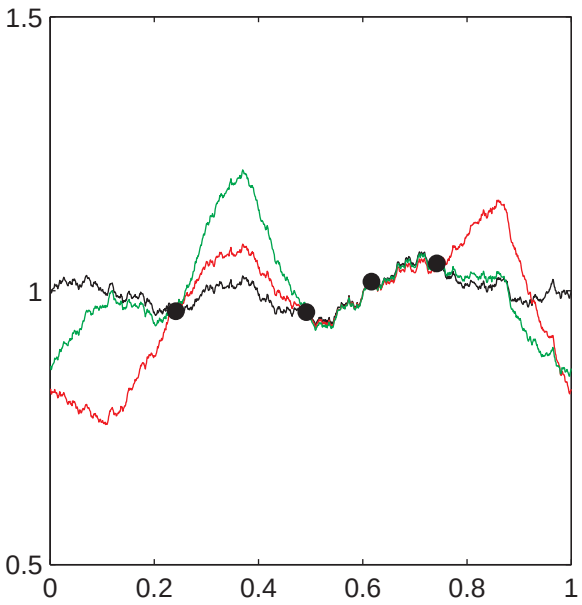


Figure 9: Non-minimum interpolating random curves (red and green ones) versus the minimum energy interpolating random curve (in black).

- Recall that in the wavelet decomposition, the coefficients of scaling functions (i.e., α) reflect the trend, while the coefficient of wavelets (i.e., β) represent the regularity (or smoothness). To ensure that the underlying function $f(\cdot)$ is Lipschitz, one only needs to impose condition (2.40) on β. If the underlying shape is a straight line, we should have $\alpha = 0$. Recall that we examine the straightness on a surface, which leads to a small value of $\|\alpha\|_2$. (If the underlying target shape is not a straight line—e.g., roundness—then the geometric shape needs to be removed from the data, before applying the aforementioned minimum energy principle.)

2.3.2.3 Optimal Sampling Positions

In this section, we consider the optimal sampling positions for $\{s_\ell\}_{\ell=1}^n$. The "optimum" is defined as follows. We assume the existence of a *true* signal. A sampling strategy is equivalent to a selection of a subset of the true signal. Based on the selected subset, the aforementioned minimum energy interpolating is applied to obtain

an *estimated* signal. The *optimum* is to minimize the norm of the difference between the *truth* and the *estimated*. We find that the optimal sampling strategy (under our model) is the one that maximizes the singular values of matrix Φ_1. Maximizing singular values of Φ_1 (via choosing different subset of rows of $\mathbf{\Phi}_1$) is a hard numerical problem. We introduce a heuristic approach instead. The rest of this subsection presents details of the justification of the above.

Recall that the following denotes the complete matrix associated with the discrete wavelet transform and it is orthogonal:

$$\begin{pmatrix} \Phi_1 & \Phi_2 \\ \Phi_1^c & \Phi_2^c \end{pmatrix}. \tag{3.49}$$

Moreover, we suppose that the true surface at the sampling positions are $y = \Phi_1\alpha_0 + \Phi_2\beta_0$, i.e., α_0 and β_0 are the wavelet coefficients of the true surface. Recall in Section 2.3.2.2 where the minimum energy interpolating algorithm is applied, one needs to generate β. Let $\tilde{\beta}$ denote such a generated β. Recall we have $y = \Phi_1\alpha + \Phi_2\tilde{\beta}$. From Lemma 2.3.1, we have

$$\begin{aligned} \alpha &= \Phi_1^T(\Phi_1\Phi_1^T)^{-1}(y - \Phi_2\tilde{\beta}) \\ &= \Phi_1^T(\Phi_1\Phi_1^T)^{-1}(\Phi_1\alpha_0 + \Phi_2\beta_0 - \Phi_2\tilde{\beta}) \\ &= \Phi_1^T(\Phi_1\Phi_1^T)^{-1}\Phi_1\alpha_0 + \Phi_1^T(\Phi_1\Phi_1^T)^{-1}\Phi_2(\beta_0 - \tilde{\beta}). \end{aligned}$$

Let $y^c = \Phi_1^c\alpha + \Phi_2^c\tilde{\beta}$. Substituting the above α, we have

$$\begin{aligned} y^c &= \Phi_1^c\Phi_1^T(\Phi_1\Phi_1^T)^{-1}\Phi_1\alpha_0 + \Phi_1^c\Phi_1^T(\Phi_1\Phi_1^T)^{-1}\Phi_2(\beta_0 - \tilde{\beta}) + \Phi_2^c\tilde{\beta} \\ &= \Phi_1^c\Phi_1^T(\Phi_1\Phi_1^T)^{-1}\Phi_1\alpha_0 - \Phi_2^c\Phi_2(\Phi_1\Phi_1^T)^{-1}\Phi_2(\beta_0 - \tilde{\beta}) + \Phi_2^c\tilde{\beta}. \end{aligned}$$

The second equation above is based on the following:

$$0 = \begin{pmatrix} \Phi_1^c & \Phi_2^c \end{pmatrix}\begin{pmatrix} \Phi_1^T \\ \Phi_2^T \end{pmatrix} = \Phi_1^c\Phi_1^T + \Phi_2^c\Phi_2^T.$$

The above is true because the matrix in (3.49) is orthogonal. In addition, we have the difference between the interpolated and the truth as

$$\begin{aligned} & y^c - (\Phi_1^c \alpha_0 + \Phi_2^c \beta_0) \\ = \; & \Phi_1^c \left[\Phi_1^T (\Phi_1 \Phi_1^T)^{-1} \Phi_1 - I \right] \alpha_0 + \Phi_2^c \left[\Phi_2^T (\Phi_1 \Phi_1^T)^{-1} \Phi_2 + I \right] (\tilde{\beta} - \beta_0). \end{aligned}$$

We consider the quantity $\|y^c - (\Phi_1^c \alpha_0 + \Phi_2^c \beta_0)\|_2^2$, which is the norm of the above difference. It will be desirable if this quantity is small. Given the above equation, recalling $n \ll N$, matrix (Φ_1^c, Φ_2^c) is a big proportion of the orthogonal matrix in (3.49). Hence the value of $\|y^c - (\Phi_1^c \alpha_0 + \Phi_2^c \beta_0)\|_2^2$ is minimized when the norm of coefficients $M_1 \alpha_0$ and $M_2(\tilde{\beta} - \beta_0)$ are minimized, where $M_1 = \Phi_1^T (\Phi_1 \Phi_1^T)^{-1} \Phi_1 - I$ and $M_2 = \Phi_2^T (\Phi_1 \Phi_1^T)^{-1} \Phi_2 + I$. Since α_0 and $(\tilde{\beta} - \beta_0)$ are prefixed, to minimize the norm of the coefficients, we need to minimize the eigenvalues of M_1 and M_2.

M_1 is a projection matrix. The eigenvalues of M_1 are 0's and 1's. Its eigenvalues are minimized (in fact, reduces to zero matrix) when Φ_1 is of full column rank. Recall Φ_1 has 2^L columns. If the row rank of Φ_1 is k, the multiplicity of one in $\Phi_1^T (\Phi_1 \Phi_1^T)^{-1} \Phi_1 - I$ is $2^L - k$. Apparently, larger k is more desirable. The maximal possible k is the sample size; i.e., Φ_1 has full row rank.

It is more delicate to study the eigenvalues of M_2. Recall that $\Phi_1 \Phi_1^T + \Phi_2 \Phi_2^T = I$, hence $\Phi_1 \Phi_1^T$ and $\Phi_2 \Phi_2^T$ can be diagonalized simultaneously. The singular value decompositions of Φ_1 and Φ_2 consequently can be written as

$$\Phi_1 = U D_1 V_1, \quad \Phi_2 = U D_2 V_2,$$

where U, V_1, and V_2 are orthogonal and D_1, D_2 are diagonal. Moreover, we must have

$D_1^2 + D_2^2 = I$. We have the following:

$$\begin{aligned} M_2 &= \Phi_2^T(\Phi_1\Phi_1^T)^{-1}\Phi_2 + I \\ &= V_2^T D_2 U^T (U D_1^2 U^T)^{-1} U D_2 V_2 + I \\ &= V_2^T (D_2 D_1^{-2} D_2 + I) V_2 \\ &= V_2^T (D_1^{-2}) V_2. \end{aligned}$$

Hence minimizing the eigenvalues of M_2 is equivalent to maximizing the eigenvalues of $\Phi_1\Phi_1^T$, which is a hard numerical question.

We consider a heuristic approach. Fig. 10 presents eight scaling functions, cor-

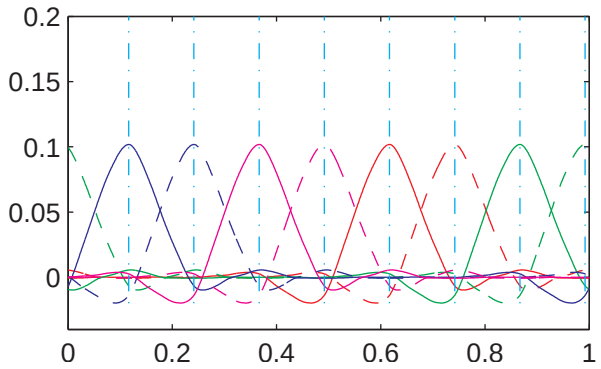


Figure 10: Scaling functions in the wavelet decomposition. The corresponding maximum positions are $0.117, 0.242, 0.367, 0.492, 0.617, 0.742, 0.867$, and 0.992.

responding to Symmlet with 6 vanishing moments at the coarsest level $L = 3$. Each x-coordinate corresponds to a row in the system (3.44). Intuitively, eigenvalues of matrix $\Phi_1\Phi_1^T$ is maximized if the matrix Φ_1 is diagonally dominated: the diagonal entries in absolute value are much bigger than off-diagonal entries. In Fig. 10, this corresponds to finding locations (x-coordinate) such that one scaling function takes big value, while the other scaling functions take values close to zero at the same site. Eight of these positions are marked by dash-doted vertical lines in the figure; they are the optimal sampling positions.

2.3.3 Form Error Assessment

Tolerances are frequently specified as maximum permissible error in form. After obtaining the baseline $b(x)$ and the half-width w of the confidence band, one can assess the form error by applying the minimum zone (MZ) method. The MZ method can find the narrowest tube that contains the confidence band; then the width of the narrowest tube is the estimate of the form error. Evidently, such approach leads to conservatism—the confidence band tends to be larger than a tube based on CMM measurement points alone. On the other hand, there has been an interest in the literature to estimate the underlying form error *unbiasedly* (e.g., [56]). In this section, we propose a method to estimate the form error that is consistent with this line of research. Simulations (in Section 2.5.2) render satisfactory results.

Recall our statistical model for the dense set of measurements:

$$Y_\ell = \sum_{j\in\mathcal{I}_L} \alpha_j \phi_j(\frac{\ell}{N}) + \sum_{i>L}^{L'} \sum_{j\in\mathcal{I}_i} \beta_{ij}\psi_{ij}(\frac{\ell}{N}) + \varepsilon_\ell, \qquad \ell = 1, 2, \ldots, N.$$

Note that in the above equation, comparing to (3.8), the measurement errors ε_ℓ are considered. We assume that $\{\frac{\ell}{N}\}_{\ell=1}^N$ are dense. Following a tradition (e.g. [56]), the 'true' form error is the outcome by applying the MZ method to $\{(\frac{\ell}{N}, Y_\ell)\}_{\ell=1}^N$. Our objective is to estimate this true form error unbiasedly using only a small subset of $\{(\frac{\ell}{N}, Y_\ell)\}_{\ell=1}^N$, i.e., using $\{(s_\ell, y_\ell)\}_{\ell=1}^n$, where $n \ll N$. Note that the positions of $\{s_\ell\}_{\ell=1}^n$ can be decided using the sampling strategy proposed in Section 2.3.2.3. Considering the measurement errors, the measurements at $\{s_\ell\}_{\ell=1}^n$ are

$$y_\ell = \sum_{j\in\mathcal{I}_L} \alpha_j \phi_j(s_\ell) + \sum_{i>L}^{L'} \sum_{j\in\mathcal{I}_i} \beta_{ij}\psi_{ij}(s_\ell) + \varepsilon_\ell, \qquad \ell = 1, 2, \ldots, n.$$

We propose the following method to estimate the form error using the measurements $\{(s_\ell, y_\ell)\}_{\ell=1}^n$.

1. Given $\{(s_\ell, y_\ell)\}_{\ell=1}^n$, we estimate the baseline sequence (denoted by $\{b(\frac{\ell}{N})\}_{\ell=1}^N$) via the minimum energy interpolating scheme proposed in Section 2.3.2.2. Given

$\{(s_\ell, y_\ell)\}_{\ell=1}^n$, after assigning $\beta_{ij} = 0$, one can compute the α as in (3.48). Consequently, the baseline $b(\frac{\ell}{N})$ can be constructed by substituting the above mentioned α and $\beta = 0$ into (3.44).

2. We generate surrogates of the sequence $\{Y_\ell\}_{\ell=1}^N$ and estimate their form error via the following:

   (a) Given the estimated A and σ, we adopt the model $\beta_{ij} \sim A \cdot 2^{-\sigma(i-L)} \cdot N(0,1)$ and generate a sequence $u(\frac{\ell}{N}), \ell = 1, 2, \ldots, N$ by applying (3.44) with the above β_{ij}'s and $\alpha = 0$. Let $u^{(k)}(\frac{\ell}{N}), 1 \leq \ell \leq N$ denote the kth generated random sequence.

   (b) Define $Y^{(k)}(\frac{\ell}{N}) = b(\frac{\ell}{N}) + u^{(k)}(\frac{\ell}{N}), \ell = 1, 2, \ldots, N$, where $b(\cdot)$ is generated in step 1, $u^{(k)}(\cdot)$ is generated in step 2(a). $Y^{(k)}(\cdot)$ is the kth surrogate sequence.

   (c) Use the MZ method to estimate the form error of $(\frac{\ell}{N}, Y^{(k)}(\frac{\ell}{N})), \ell = 1, 2, \ldots, N$, and denote it by e_k.

3. Repeat the step 2 for large enough k, and our final form error estimate is the ν-quantile of the e_k's. Specification on how to decide ν is given in Appendix A.11.

Note that both confidence band and form error can be used in quality assessment. We consider them as complimentary methods. Simulations will be performed to study their properties.

2.4 Real Data Study

The assumptions required for our method are verified with the real CMM data in Section 2.4.1. In Section 2.4.2, the confidence bands for the data are constructed using the proposed method.

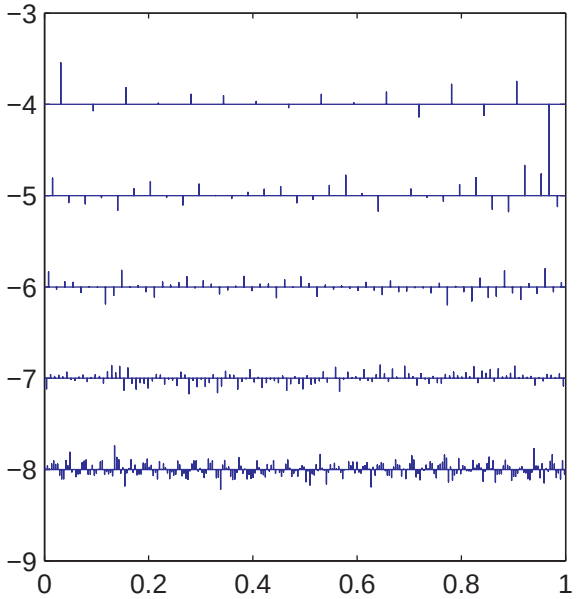
2.4.1 Assumption Verification

We consider data taken both by point-by-point and by scanning. They are plotted in Fig. 7. The wavelet transform is carried out. We choose a nearly symmetric wavelet (Symmlet) with six vanishing moments. For each sequence, we extract out a length-512 subsequence. Fig. 11 is a standard way to display wavelet coefficients.

Recall (2.40) is our key assumption. (When (2.40) holds, by Theorem 2.2.2, the underlying function $f(x)$ is Lipschitz.) If (2.40) holds, the maximum absolute value of wavelet coefficients as a function of the scale i must be below a decreasing straight line. In Fig. 12, we plot the maximum absolute value of wavelet coefficients at all scales. We can clearly see the linear pattern for the scanning data. This experiment has been repeated for different segments of the original signal. The above pattern has been consistently observed. The corresponding figure for the point-by-point case (Fig. 12 (a)) does not show such a pattern. The likely cause of this difference is the errors associated with repositioning the probe during the point-by-point measurement. We choose the scanning data to examine and measure the smoothness of the surface.

We study the distribution of the wavelet coefficients at a fixed scale. In Fig. 13, the QQ-plots of the wavelet coefficients at the same scale are plotted. The x-coordinates are the standard normal quantiles, while the y-coordinates correspond to the quantiles of the wavelet coefficients at a fixed scale. For illustration purpose, all coefficients are multiplied by 10^4. Most of the QQ-plots indicate a fit to the normal distributions, except for scale 5 in both cases. For more quantitative results, the Jarque-Bera hypothesis test is run and the corresponding p-values are 0.001, 0.3716, LT5, and LT5 for the four scales in the point-by-point case. (Here LT5 stands for larger than 0.5.) The p-values for the four cases in the scanning measurements are 0.1118, LT5, LT5, and 0.484981. Note the latter case is more interesting, because we study the underlying surface, and the scanning CMM measurements are more faithful to the smoothness of the true surface. The p-values at scale 5 are small (in particular

(a) Wavelet Coefficients for the Point-by-Point CMM Measurements



(b) Wavelet Coefficients for the CMM Measurements via Scanning

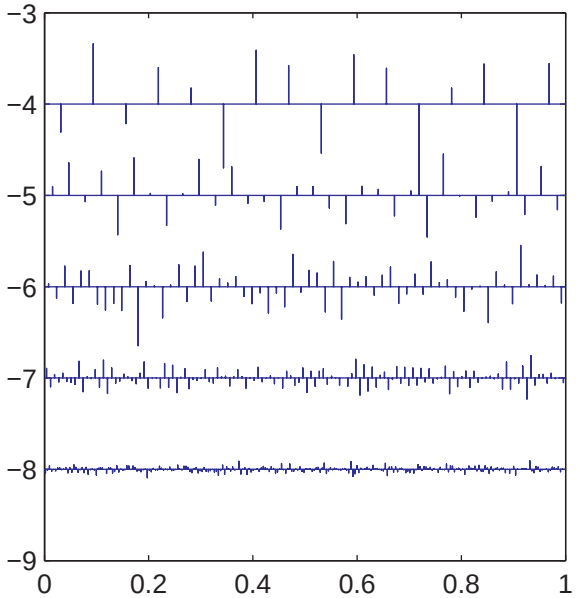


Figure 11: Wavelets Coefficients. Wavelet coefficients at the same *scale* are plotted on the same horizontal line. All wavelet coefficients are scaled—the stick length is proportional to its absolute value. The horizontal axis represents the *location*.

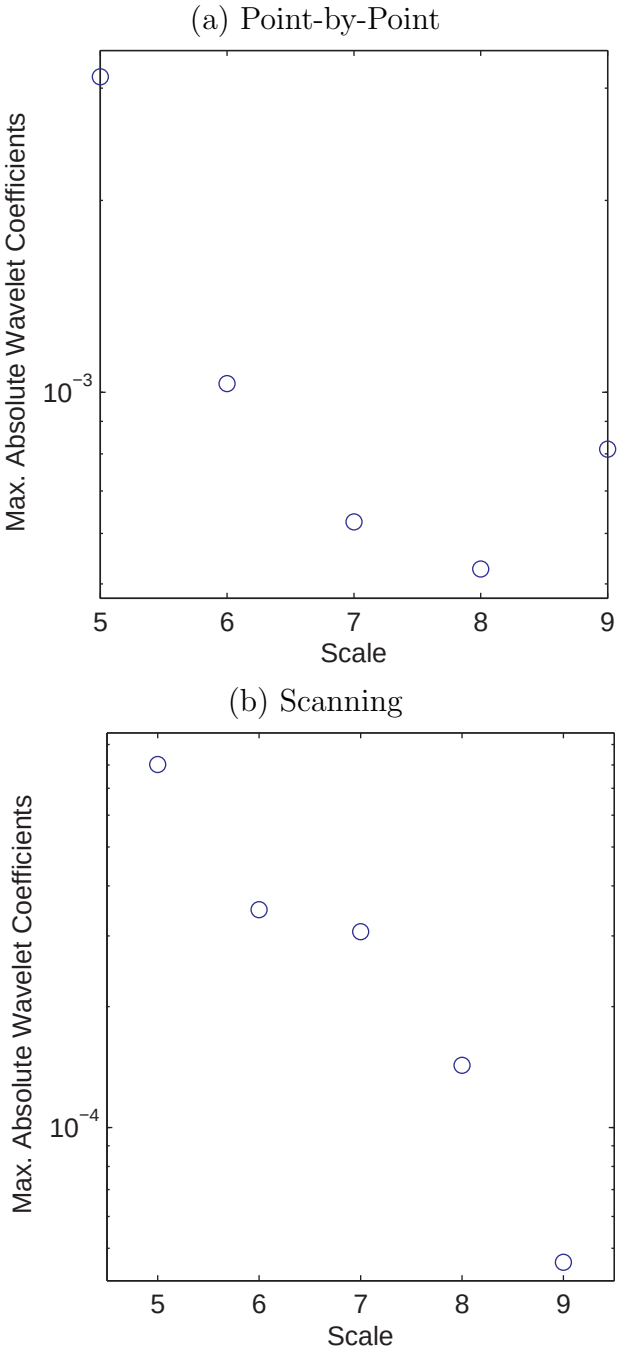


Figure 12: The maximal absolute values of wavelet coefficients per scales versus the scales. The horizontal axis represents the scale. The vertical axis reflects the logarithmic transformed maximum absolute value of wavelet coefficients at each scale. In (b), we observe a decreasing line pattern, which is consistent with (2.40).

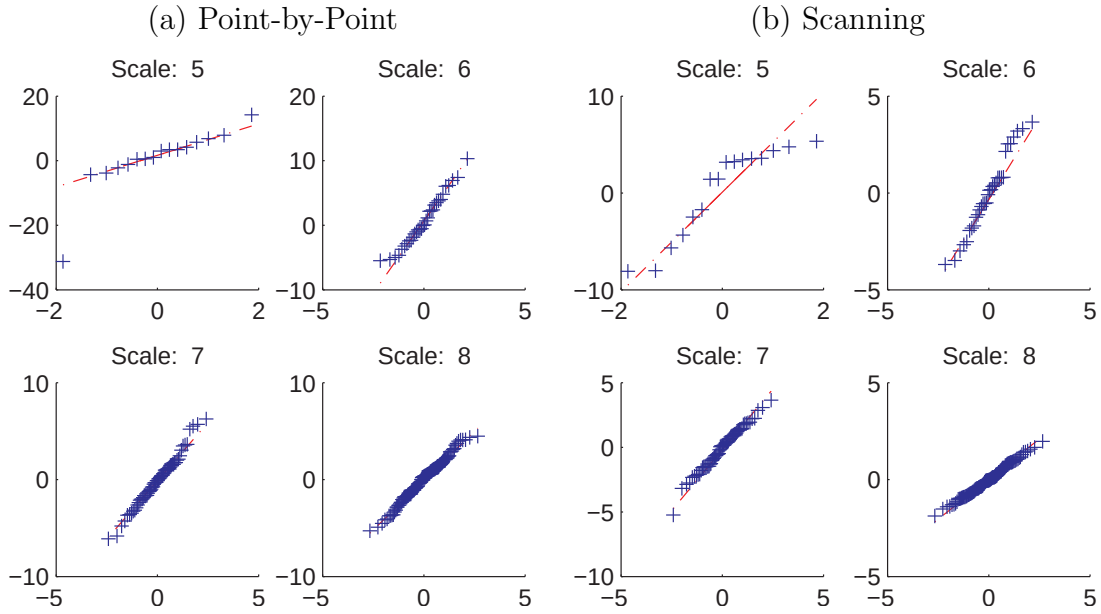


Figure 13: The QQ-plots of the wavelet coefficients at all scales. Each subfigure includes wavelet coefficients at a particular scale.

when the point-by-point measurements are considered); however in our framework, the finer scale (i.e., when the scale index is large) is of more interest, because in (2.41), one can increase the value of L by 1, so that the part of the signal associated with scale 5 becomes the part of the signal expressed by the scaling functions (i.e., the first term on the right hand size of (2.41)). In summary, it is reasonable to assume that the wavelet coefficients (β_{ij}) satisfy the normal distribution.

2.4.2 Confidence Bands for CMM Data

We construct the confidence bands for the two CMM data sets that are displayed in Fig. 7. Recall that we assume $\beta_{ij} \sim A \cdot 2^{-\sigma(i-L)} \cdot N(0,1)$. We need to estimate A and σ, which correspond to the intercept and slope in Fig. 12 (b). We apply simple linear regression to obtain the estimates: $\hat{\sigma} = 0.69$ and $\hat{A} = 0.675 \times 10^{-3}$. Applying these parameters, we can determine the half-width of the confidence band via the simulation approach in Section 2.3.1.2. We found the empirical percentiles (based on 10^6 simulated sequences) corresponding to 99%, 99.5%, and 99.9% are 5.84×10^{-4},

6.09×10^{-4}, and 6.62×10^{-4}, respectively. The baseline can be estimated via the aforementioned wavelet shrinkage method. The level 99% and 99.9% confidence bands for each case are displayed in Fig. 14.

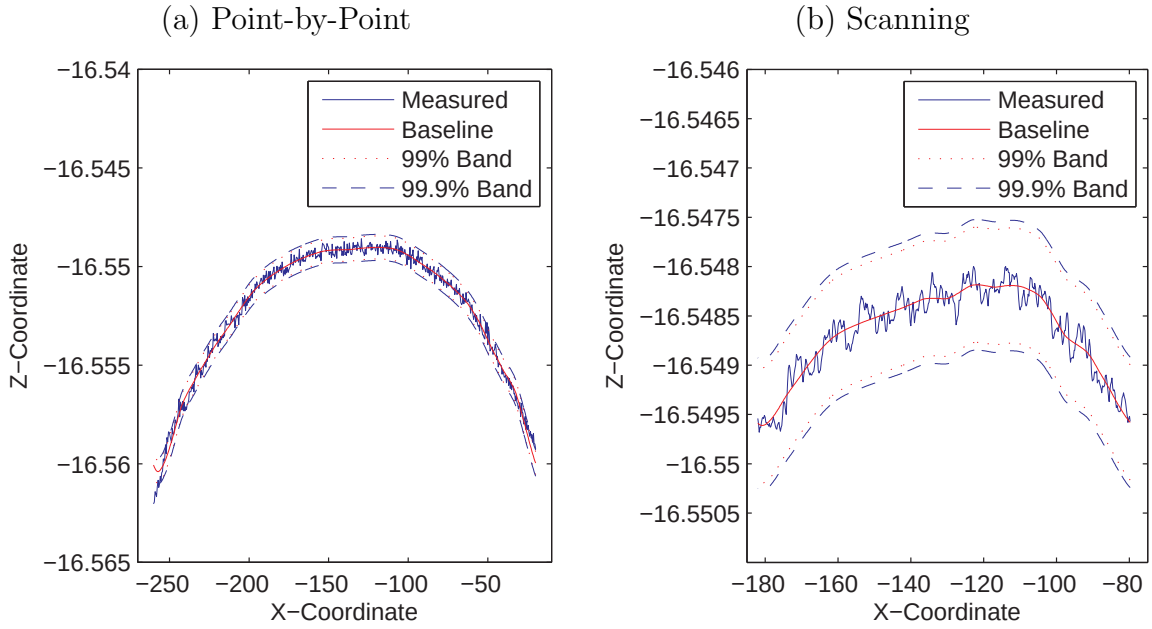


Figure 14: The 99% and 99.9% confidence bands for the CMM measurement displayed in Fig. 7. Note that the band is for the underlying surface, not the CMM measurements. The fact that the band barely covers all CMM measurements in (a) may be caused by the large repositioning noise in the point-by-point scheme.

2.5 Simulation Study

We study the confidence band in Section 2.5.1. The form error estimate is evaluated in Section 2.5.2.

2.5.1 Simulations on Confidence Bands

In this section, we use synthetic data to examine our method. We will gain insights on our sampling strategy. We assume that the CMM measurement sequence satisfies the model that is proposed in Section 2.2.4. We set $\sigma = A = 1$. For implementation convenience, we choose the length of sequence $N = 512$. By doing so, we can take advantage of the existing implementation of the discrete wavelet

transforms in WaveLab (`http://www-stat.stanford.edu/~wavelab/`). We choose the coarsest level to be $L = 4$. (The choice of L allows subjectivity, which does not hurt the foundation of our model.) As mentioned in model description, we choose Symmlet with six vanishing moments. With these conditions, as discussed in Section 2.3.2.3, the optimal sampling positions with 16 ($= 2^L$) points should be $27, 59, 91, 123, 155, 187, 219, 251, 283, 315, 347, 379$, 411, 443, 475, and 507. Note that the positions may vary due to differences in the implementation of the discrete wavelet transform. The above values are based on the implementation in WaveLab.

After taking the measurements at the optimal sampling positions, we estimate the corresponding baseline using the proposed interpolating algorithm in Section 2.3.2.2. When applying (3.48), we set $\beta = 0$ (i.e., all fine scale wavelet coefficients are set to be zero) so that the resulting baseline is smooth. We then select the half-width of the confidence band according to the description in Section 2.3.1.2. Note that with the aforementioned choice of A and σ, the sample percentiles based on one millions simulations are 0.563925(99%), 0.589235(99.5%), and 0.645583(99.9%). There is a caveat in using these quantiles. Note that the resulting confidence band makes sense if the coefficients associated with all scaling functions (i.e., α) are known. Such an assumption is not realistic. However, in simulation, one can estimate the pointwise standard deviation of the CMM measurements, using the pointwise standard deviation between the estimated baseline and the simulated sequence. In our case, we found that the estimated pointwise standard deviation ≈ 0.2. So 0.2 is added to the above quantiles. The reason we choose 0.2, instead of e.g., 3×0.2, is that the simulations show the latter is too conservative. More analytical study here is possible to make the width selection more accurate. Fig. 15 illustrates one scenario of the above procedure.

To validate our proposed confidence band method and our choice of the sampling positions, we carry out numerical studies in the above mentioned framework. Three

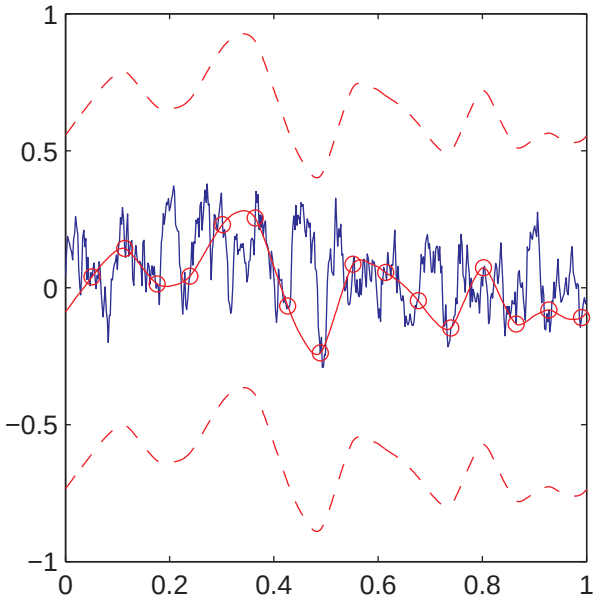


Figure 15: A synthetic example. The noisy curve is simulated data, based on the properties of CMM measurements. Circles indicate positions where measurements are obtained. Two dashed curves circumscribe the 99.9 % confidence band. The fact that the confidence band includes all CMM measurements demonstrates the success of the band.

cases are examined:

(C1) All $2^L = 16$ positions are at the optimal sampling positions that are derived in Section 2.3.2.3.

(C2) Only part of the previous 16 positions are taken: the 5th, 7th, 9th, and 11th positions are removed. We would like to study the case when the number of samples is less than 2^L, i.e., $n < 2^L$.

(C3) Still take 16 sampling positions, however they are not the optimal sampling positions that are specified in Section 2.3.2.3. We did so by shifting all positions in (C1) to the left by 16. We can also use random sampling positions. Similar results are observed.

Ten thousand experiments are run for each of the above three cases. The number of times that the constructed 99.9 % confidence bands do *not* cover the generated

CMM sequences are $100, 3906$, and 5661 for C1 to C3, respectively. We learned the following lessons:

1. Recall one objective is to construct a confidence band (which bases on a small number of measurements), such that this band contains the 'true' surface with high probability. We do observe that the number of uncovered cases is the minimum in C1. This result validates our construction of the confidence band.

2. The number of uncovered cases in C2 and C3 are significantly larger than the corresponding number in C1. It demonstrates the optimality of our new sampling strategy.

The above leads to the following guideline in adopting our model in practice: making the experimental design conform with C1. For most CMMs, it is not hard to implement this guideline.

2.5.2 Comparison with Traditional Methods

In this subsection, we estimate form error as described in Section 2.3.3, and compare the results with the two traditional methods: the Minimum Zone (MZ) method and the Orthogonal Least Squares (OLS) method. For the comparison, we focus on the unbiasedness of the form error estimates (similar to [56]). We calculate the ratios of the estimated form errors over the true form errors; note that the estimate is less biased if the calculated ratio is closer to one. The specific procedure is as follows.

Step 1. Simulate a dense enough set of measurements. Specifically, set $N = 512$, $\sigma = A = 1$, $L = 4$, and generate the function as $y = \mathbf{\Phi}_1\alpha + \mathbf{\Phi}_2\beta$, where $\beta_{ij} \sim A \cdot 2^{-\sigma(i-L)} \cdot N(0,1)$ and $\alpha_j = 0, \forall j$.

Step 2. Specify a potential tolerance τ for part acceptance. For this, generate 1000 functions as in Step 1. Let m_i denote the maximum magnitude in the ith function. We choose τ as $\tau = 1.5 \times \max_i m_i$.

Step 3. Add measurement errors to y in Step 1. That is, $y = \mathbf{\Phi}_1\alpha + \mathbf{\Phi}_2\beta + \epsilon \cdot N(0,1)$, where ϵ incorporates the errors of individual CMM measurement. We choose $\epsilon = \tau/5$ (It is a reasonable requirement that the measurement equipment uncertainty must be 5 times smaller than the tolerance zone). Determine the form error using the MZ method, and treat it as the "true" form error.

Step 4. From the measurements in Step 3, take the small number of samples using the sampling approach C1 $\sim$ C3.

Step 5. Using the samples taken in Step 4, estimate form error using the three different methods (MZ, OLS, and our method), and calculate the ratios of the estimated form errors over the true form error.

Table 4 compares the calculated ratios from the three methods. The values in parentheses are standard errors. Both MZ and OLS underestimate the form error. Our method is less biased. [56] compares Gaussian process (GP) models with MZ and OLS as well. By comparing to their results, we believe that GP and our method will render similar performance. Table 4 reconfirms that C1 is the best sampling strategy under our model.

Table 4: MZ, OLS, and our method are compared for the unbiasedness of the form error estimation. The averaged ratios based on 1000 simulations are shown. The standard errors are in the parentheses.

	MZ	OLS	Our method
C1	0.5618 (0.1279)	0.5912 (0.1370)	0.9904 (0.1287)
C2	0.5084 (0.1353)	0.5350 (0.1432)	0.9378 (0.1252)
C3	0.5889 (0.1245)	0.6184 (0.1315)	1.7426 (0.4507)

2.6 Discussion

We aim at developing a general theory on the construction of the confidence band, form error assessment, and the sampling strategy. In testing the measurement compliance with the requirements and determining sampling positions, none of the existing works considers the wavelet-based model. We are the first to utilize Lipschitz property to establish a statistical model for CMM measurements. Our data-driven approach to identify a statistical model represents a trend in modern functional data analysis. The proposed work differs from existing basis function approaches (which may uses wavelets as basis functions): we use the wavelet framework as an intermediate tool to create a statistical model for Lipschitz functions, while most existing work uses wavelets as a nonparametric smoothing tool. In the proposed model, specific types of wavelets (i.e., not all types of wavelets) must be adopted; our treatment of wavelet coefficients is different from other methods when wavelets are merely used for smoothing.

Extension to 2*-D.* So far, we have discussed the case when the underlying boundary is a 1-D function f residing in the unit interval $[0, 1]$. It can be straightforwardly extended to 2-D surface, if we can assume that the 2-D surface is a tensor production of two 1-D functions; although in reality, such an assumption may be too optimistic. Tensor production simply implies that $h(x, y) = f(x) \cdot g(y), x, y \in [0, 1]$, where $f(x)$ and $g(y)$ are uniformly Lipschitz $\gamma > 0$. The assumption of tensor production really simplifies the derivation. Most of the algorithm in 2-D is nearly parallel with the algorithm in 1-D. However, 2-D is a much more delicate problem than 1-D. See related works: [15], [7], [18]. We need more research on this topic.

We will extend the present model for other geometric shapes. To do so, one may subtract an estimated form from the measurements, so that the problem is converted into a straightness problem. Due to space, this paper does not explore further in this direction.

2.7 Conclusion

We propose a wavelet-based method to construct confidence bands, to assess form errors, and to determine an optimal sampling strategy for CMM measurements in coordinate metrology. The confidence band for the measurement data is computed via specifying its baseline and the half-width of the confidence band; the former can be constructed via the wavelet shrinkage method, and the latter can be obtained via simulation based on the Lipschitz regularity. For the form error assessment, we predict the surface of the geometric feature (a dense set of measurements) using a small number of observed samples, and find the maximum inscribing and minimum circumscribing geometry that bounds all points on the predicted surface. For the optimal sampling strategy, a wavelet-based random curve interpolating algorithm is considered. We discuss the optimal choice of new sampling positions under our model. The proposed method has been validated with synthetic and real data.

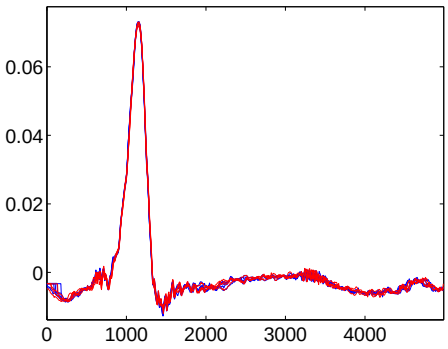
CHAPTER III

A SINGLE INTERVAL BASED CLASSIFIER

3.1 Introduction

We study a class of classification problems, in which it is known *a priori* that there exists an interval, such that a classifier built on this interval can achieve the best possible performance. An example can be seen in the tonnage data analysis [27]. Fig. 16 plots 73 curves. Each curve is 4990 dimensional. Among these curves, 7 of them are considered faulty (in red) and 66 are considered normal (in blue). In Fig. 16 (a), it is nearly impossible to distinguish these two classes. On the other hand, it is known that the two classes are well separated by segments restricted on the time interval $[810, 845]$. To classify faulty and normal curves, it is desirable to build a classifier that is only based on the aforementioned interval. Such an interval will be called an *optimal interval*. A classifier restricted on the optimal interval leads to faster and more effective prediction on new curves. A challenge is: we may know the existence of the optimal interval, however not the exact location and size of it.

(a) Entire Curves



(b) Within the Optimal Interval

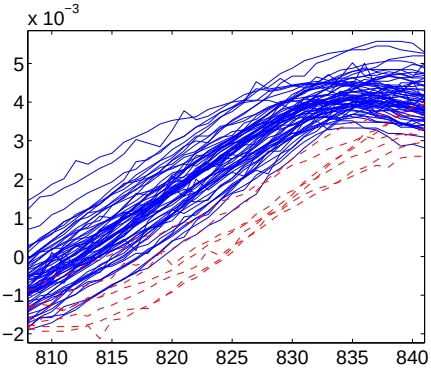


Figure 16: The curves are 4990 dimensional. There are 66 normal curves and 7 faulty curves. Left figure shows that a comparison based on the entire curves is insignificant. Right figure shows that the curves can be clearly separated over the interval $[810, 845]$.

The aforementioned tonnage data come from monitoring a transfer or progressive die process, which performs multiple operations by means of a die having several stations, each of which performs a different operation as the stock passes through the die. Fig. 17 depicts a transfer die process that has six stations to produce a doorknob part. Generally, the press tonnage signal measurement is performed by

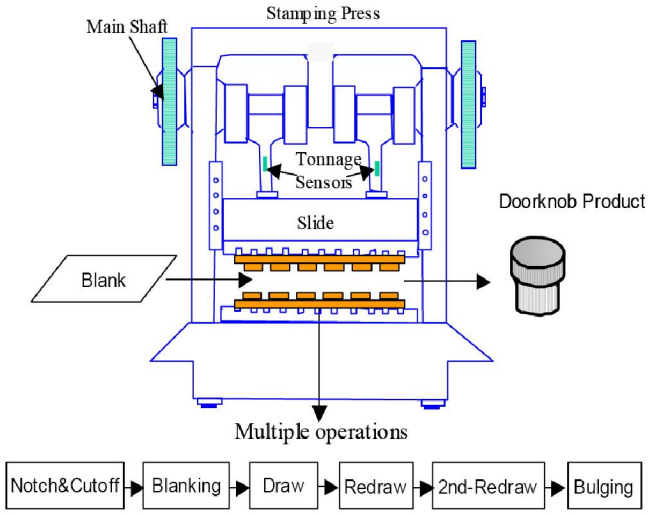


Figure 17: Doorknob transfer die process.

the tonnage sensors (strain gage sensors) installed on the press linkages or columns. As an example of the doorknob process, two tonnage sensors are installed on the press linkages as shown in Fig. 17. The obtained press tonnage signal represents the total stamping force applied to all working stations in the press. Press tonnage signals can be continuously collected during production by a computer-based tonnage monitoring system. To monitor the condition of each working station, it is required to know the signal profile of the stamping force at each station. Methodology has been developed [27, 26] to decompose the total stamping force into the individual forces generated from each station. For a given tonnage signal, it naturally breaks into various signal segments with each of them corresponding to specific stamping machine,

die, and part interactions or conditions. Under normal production conditions, the tonnage signals are repeatable, or belong to the same baseline, for a given part in a fixed production condition. However, if some failures occur in production (machine failure, die broken or worn out, material properties change, part quality variation, etc.), the signals in some segments of the tonnage signals may change. The specific segments and their corresponding changes are different as the types of failures differ. However, the signals under given failure conditions are still repeatable, but belong to a different baseline that deviates from the normal conditions (as well as other failure conditions). In a production environment, because the tonnage signals are continually collected under various (unknown) system conditions, it is important to perform automatic classification to help grouping the tonnage signals into different clusters, which correspond to different production conditions. In the meantime, the optimal segments should be automatically determined to improve the sensitivity and performance of the classification.

We propose a classifier, which is built on an estimated optimal interval. We name our method an *interval based classifier* (IBC). The proposed method has three stages.

1. We estimate the *baseline* for each class by minimizing an objective function that is proportional to a weighted L_1 norm. Justification of such a criteria is provided. This step can be solved efficiently via standard convex optimization—a key component of operations research.

2. The optimal interval for classification is identified. We study the numerical details and demonstrate that it can be achieved via a low-order-of-complexity algorithm. Our analysis is based on studying the Lagrangian multiplier version of the original problem. The proposed method is fast: it takes a few minutes on a laptop to find the optimal classifier for a data set having nearly 100 curves and dimensionality of 4990.

3. A classifier that is based on the previously estimated optimal interval is proposed.

Our method is applied to a real data set from stamping processes. The outcome not only confirms our findings, but also improves our understanding of the data, which used to be rooted in the physical knowledge of the processes.

In this chapter, we formulate a special classification problem from data mining into an optimization problem, and solve the problem with analytical study and demonstrate with real data. This chapter makes several contributions. First of all, to the best of our knowledge, this is the first time that an interval-based classifier is proposed. Secondly, our method should be applicable in situations when features are interval based. Lastly, our derivation of the algorithm in the second stage generates a surprise, which consequently leads to an advantage of the IBC: it is easy to choose the algorithmic parameter in IBC. We explain more in the remaining of this paragraph. Recall IBC requires locating the optimal interval, which can be of any length. Note that when the dimension of the curve is large, one could have a large number of possibilities to examine. We found that one only needs to study those lengths that are associated with the vertices of a convex hull of an objective function. They, in our numerical experiments, form about 1% of all possible lengths. Hence IBC in practice can be done very efficiently—much more efficient than it appears. The above will become more evident in our description of numerical experiments.

We compare IBC with a state-of-the-art classifier—fused lasso [50]—which utilizes stepwise constant baselines. The original goal of the fused lasso is to produce interval-like solutions, given that the coefficients are sparse or piecewise constant. (The major difference between IBC and fused lasso is that IBC is a classifier that is based on an interval, while fused lasso is a classifier whose coefficients resemble a step function—i.e., piecewise constant.) The fused lasso is a combination of lasso [49] and variable fusion [28], and it penalizes both the coefficients and their successive differences. The

fused lasso provides piecewise solution, however it is *not* guaranteed to generate the unique optimal interval, which is achieved in IBC. In particular in simulations, we run a comparison between IBC and an extension of the fused lasso—fused lasso support vector classifier. We demonstrate that for a real data set, the two methods produce similar solutions, but IBC is more stable and guarantees the unique optimal interval.

The rest of the chapter is organized as follows. The IBC is described in Section 3.2. Numerical experiments in Section 3.3 demonstrate the effectiveness of IBC for both synthetic and real data. We conclude in Section 3.4.

3.2 Interval Based Classifier (IBC)

We give an overview of the proposed method for classification based on the most influential segment. In the remaining of this chapter, we consider the "most influential segment" interchangeable with the "optimal interval." The former is more appropriate to be used when discrete algorithms are described; the latter is used when the problem is analyzed in continuum. Section 3.2.1 presents an estimator of the baseline of each class. Section 3.2.2 considers how to find the most influential segment with a prescribed size of the segment. Section 3.2.3 gives the main algorithm to find the all possible optimal intervals. Some physical justifications are provided in Section 3.2.4. Our classification rule is presented in Section 3.2.5. Finally, Section 3.2.6 describes a divide-and-conquer strategy, which can be used to deal with long curves.

3.2.1 Baseline Estimation

The first step is to estimate the baseline of each class. Let $c_{i,j}$ denote the jth curve in the ith class, $i = 1, 2$ and $j = 1, 2, \ldots, n_i$, where n_i denotes the number of curves in class i. For each class, let b_i denote its baseline. Let $c_{i,j}(x)$ (respectively, $b_i(x)$) denote the value of the curve (respectively, the baseline) at location x, $1 \leq x \leq N$, where N is the length of the discrete curve (respectively, baseline). For class i, we

estimate the baseline b_i by solving the following:

$$\begin{aligned} \min_{b} \quad & \sum_{j=1}^{n_i} \sum_{x=1}^{N} |(c_{i,j}(x) - b(x))/\sigma_i(x)|, \\ \text{subject to} \quad & b \in \mathcal{F}, \end{aligned} \tag{2.50}$$

where $\sigma_i(x)$ denotes an estimated standard deviation of errors in $c_{i,j}$'s at location x, $\mathcal{F}$ denotes a set of sequences that satisfies the following:

$$\mathcal{F} = \{f : \sum_{x=1}^{N-1} |f(x+1) - f(x)| < C_1 \text{ and } \sum_{x=1}^{N-2} |f(x+2) - 2f(x+1) + f(x)| < C_2\},$$

where C_1 and C_2 are predefined constants. There is a natural choice of C_1 and C_2: we compute the same quantities for the observed curves; then C_1 and C_2 are a constant (e.g., 1.5, to accommodate more functions) multiplied with the computed values. As for more general methods, cross validation (CV) can be also used for the choice of C_1 and C_2. With the above specification, the optimization problem in (2.50) has a convex objective function and convex constraints. Efficient solvers are available. We used a general-purpose optimization toolbox: cvx [20]. The execution time is quite impressive, referring to the section on numerical experiments.

3.2.2 Finding the Most Influential Segment with a Prescribed Length

For a fixed length s of the target segment, we consider how to compute the most influential segment. Let $\hat{b}_1$ and $\hat{b}_2$ denote the estimates for b_1 and b_2. For a given length of the segment s, the most influential segment is the segment I^* that solves the following problem:

$$\max_{1 \le i \le N-s+1} \quad \left| \sum_{x=i}^{i+s-1} \frac{\hat{b}_1(x) - \hat{b}_2(x)}{\sigma(x)} \right|, \tag{2.51}$$

where $\sigma(x)$ is a pooled estimate of the standard deviation of errors at location x. If i^* is the maximizer of (2.51), we have $I^* = [i^*, i^* + 1, \ldots, i^* + s - 1]$. For simplicity, we denote the maximum value of the objective function in (2.51) as $h(s)$. The following theorem describes the complexity of solving problem (2.51). Proof of the theorem has been relegated to Appendix A.12.

Theorem 3.2.1 *For a fixed segment length s, solving (2.51) can be done via an $O(N)$ algorithm.*

3.2.3 Convex Hull and Admissible Lengths

If we try to solve for $h(s)$ for every $s(1 \leq s \leq N)$, the overall complexity can be $O(N^2)$. This order of complexity is too high for large N. It appears that we only need to compute $h(s)$ for a small subset of s. In particular, we only need to solve for $h(s)$ when $(s, h(s))$ is a vertex of the convex hull of the function $h(s)$. An illustration of the convex hull of function $h(s)$ is given in Fig. 18. The thick red line in the figure is a convex hull. Red circles are the vertices of the convex hull. We would argue that only the vertices of the convex hull worth considering. If $(s, h(s))$ is *not* a vertex of the convex hull, we must have s_1 and s_2, such that $s_1 < s < s_2$ and

$$h(s) < \frac{s_2 - s}{s_2 - s_1} h(s_1) + \frac{s - s_1}{s_2 - s_1} h(s_2);$$

hence the most influential segment of length s can be substituted by a linear combination of two most influential segments of lengths s_1 and s_2, without sacrificing classification performance. The following theorem is an important observation in deriving our algorithm, which will be elaborated.

Theorem 3.2.2 *If $(s^*, h(s^*))$ is a vertex of the convex hull of the function $h(s), 1 \leq s \leq N$, then there must exist a constant λ, such that*

- *Segment I^* solves for*

$$\max_I \left| \sum_{x \in I} \frac{\hat{b}_1(x) - \hat{b}_2(x)}{\sigma(x)} \right| - \lambda |I|, \tag{2.52}$$

  where $|I|$ denotes the length of a segment I.

- *We have $s^* = |I^*|$ and $h(s^*) = |\sum_{x \in I^*} \frac{\hat{b}_1(x) - \hat{b}_2(x)}{\sigma(x)}|$.*

The proof of the above theorem utilizes standard arguments associated with the Lagrangian multipliers. We decide to omit a detailed proof. Note that in Fig. 18,

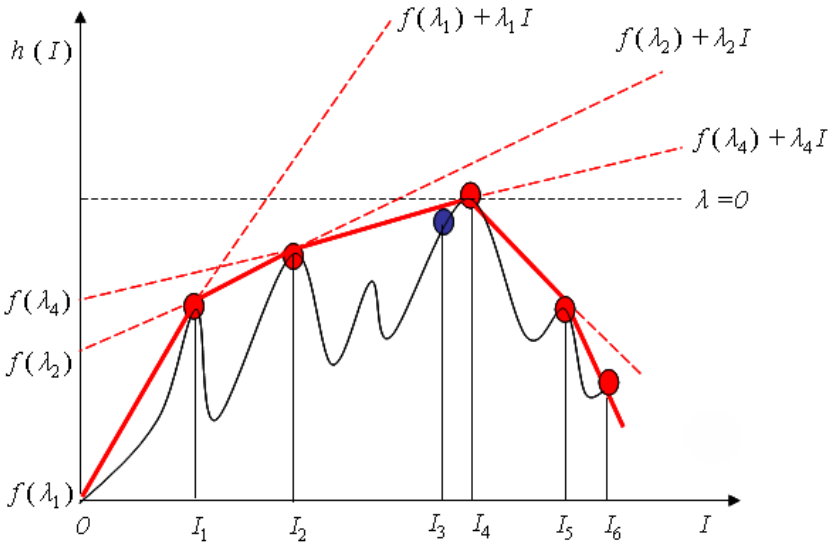


Figure 18: An illustration of the convex hull. Note that I or I_* denotes the segment lengthes. Intercept $f(\lambda)$ is defined in the context.

$f(\lambda)$ denotes the maximum intercept associated with slope λ: i.e., for the I^* and λ in the setting of (2.52), we have $f(\lambda) = h(s^*) - \lambda s^*$. Recall problem (2.51) can be solved at $O(N)$. Using the same argument, one can verify that problem (2.52) can also be solved at $O(N)$.

If $(s, h(s))$ is a vertex on the convex hull of $h(s)$, we define s an *admissible length.* It turns out that admissible lengths are a small portion of all possible lengths. For example, for the tonnage data illustrated in Section 3.3.2, there are 4990 possible lengths; however, the number of admissible lengths is just 49 (roughly 1 percent). The inadmissible length is similar to the concept of inadmissible tree sizes in another statistical modeling problem [25]. We now present a fast algorithm to search for all admissible lengths, as well as the corresponding most influential segments.

Algorithm

- **Initialization.** Take two initial values $0 < \lambda_{\min} < \lambda_{\max}$. Solve problem (2.52) with $\lambda_{\min}$ and $\lambda_{\max}$. Let s_1 and s_0 denote the lengths of the corresponding two most influential segments. It is evident that $s_0 < s_1$. (Ideally, we should have

$s_0 = 1$ and $s_1 = N$.) Readers can get an intuition from Fig. 18. Initialize: OPEN $= \{pair\ (s_0, s_1)\}$; i.e., set OPEN contains one pair.

- **While** OPEN is not empty,

  Takes a pair $(s_1', s_2'), s_1' < s_2'$, out of the set OPEN. Compute

  $$\lambda = \frac{h(s_2') - h(s_1')}{s_2' - s_1'}.$$

  Solve problem (2.52) with the above λ. Let s_3' denote the length of the corresponding most influential segment. It is easy to verify that

  $$s_1' \leq s_3' \leq s_2'.$$

  If $s_1' < s_3'$, insert pair (s_1', s_3') into OPEN.

  If $s_3' < s_2'$, insert pair (s_3', s_2') into OPEN.

  End of While.

Because solving problem (2.52) requires $O(N)$ operations, the overall complexity of the above algorithm is the number of admissible lengths times $O(N)$. We will see that the number of admissible lengths is much smaller than N. In our numerical experiments with a real data set, the number of admissible lengths is roughly one percent of N.

3.2.4 Physical Interpretation

The tonnage signal is the force measurement with x-axis being the crank angle (or equivalently as the travel distance by the upper die), and y-axis being the stamping force. The area covered under the tonnage signal is proportional to the forming energy. Under the same production conditions, the forming energy should be the same to form a designed part. The following interpretations come naturally:

- Minimizing in (2.50) is to find a baseline of a group of tonnage signals that have the similar forming energy in forming a part;

- Maximizing in (2.51) is to find the segment that can maximize the forming energy difference in the given segment. Such a segment reflects tonnage signals that are the most sensitive to the different forming conditions.

3.2.5 Classification Rule

Using the estimated baselines ($\hat{b}_1$ and $\hat{b}_2$) and identified optimal interval (I^*), a new curve can be classified. Let c_{new} denote a new curve. The classification rule of IBC works as follows: If $\sum_{x=I_s^*}^{I_e^*}(|c_{new}(x) - \hat{b}_1(x)| - |c_{new}(x) - \hat{b}_2(x)|)/\sigma(x) < 0$, where I_s^* and I_e^* are the starting and ending points of I^*, respectively, and $\sigma(x)$ is a pooled estimate of the standard deviation at location x, then c_{new} is classified into class 1; otherwise, c_{new} is classified into class 2. Note that this classifier is a *nonlinear* one, unlike a support vector classifier that will be compared with later.

3.2.6 Divide and Conquer Strategy for Long curves

The problem (2.50) of finding the baselines can be solved more efficiently by utilizing a divide-and-conquer strategy. The divide-and-conquer strategy works by dividing the original curve into several segments. Each segment of the curve is treated separately to find a baseline of the corresponding segment. All the separately estimated baselines are then concatenated to obtain the baseline of the original data. We cannot guarantee that the concatenated baseline is the same as the baseline that is estimated from the entire data. To resolve this problem, the neighboring points belonging to different segments are treated specially. Those special points are called the 'connecting-points group.'

We give a specific strategy to find the baseline with a divide-and-conquer strategy. Suppose that we divide the original data into n segments: $\{1, \cdots, p_1\}$, $\{p_1{+}1, \cdots, p_2\}$, $\cdots$, $\{p_{n-1} + 1, \cdots, p_n(= N)\}$, where p_i is the ending point of the ith segment, $i = 1, \cdots, n$. We define the ith connecting-points group as $\{p_i - 2, \cdots, p_i + 3\}$, where $i = 1, \cdots, n-1$. The divide-and-conquer strategy can be presented as follows:

1. Initialization.

   (a) For $1 \le x \le p_1 + 1$, solve the problem (2.50). Let b_1^* denote the solution.

   (b) For $i = 2, \cdots, n-1$, for $p_{i-1} \le x \le p_i + 1$, solve the problem (2.50) with the two constraints: $b_i^*(p_{i-1}) = b_{i-1}^*(p_{i-1})$ and $b_i^*(p_{i-1}+1) = b_{i-1}^*(p_{i-1}+1)$, where b_i^* denotes the solution of the ith problem.

   (c) For $p_{n-1} \le x \le p_n (= N)$, solve the problem (2.50) with the two constraints: $b_n^*(p_{n-1}) = b_{n-1}^*(p_{n-1})$ and $b_n^*(p_{n-1}+1) = b_{n-1}^*(p_{n-1}+1)$. Let b_n^* denote the solution. Note that we separate (c) from (b) because when $i = n$, $p_i + 1$ should be changed to p_i.

2. Followup Iteration. For $i = 1, \cdots, n-1$, for the ith connecting-points group, $p_i - 2 \le x \le p_i + 3$, solve the problem (2.50) with the four constraints: $b_{c,i}^*(p_i - 2) = b_i^*(p_i - 2)$, $b_{c,i}^*(p_i - 1) = b_i^*(p_i - 1)$, $b_{c,i}^*(p_{i+1} + 2) = b_{i+1}^*(p_{i+1} + 2)$, $b_{c,i}^*(p_{i+1} + 3) = b_{i+1}^*(p_{i+1} + 3)$, where $b_{c,i}$ denotes the solution to the ith connecting-points group.

3. If $b_{c,i}^*(p_i)$ and $b_{c,i}^*(p_{i+1}+1)$ are close enough to $b_i^*(p_i)$ and $b_i^*(p_{i+1}+1)$ for all $i = 1, \cdots, n-1$, terminate. Otherwise, go to step 1 and repeat the above with the additional constraints $b_i^*(p_i) = b_{c,i}^*(p_i)$ and $b_i^*(p_i + 1) = b_{c,i}^*(p_i + 1)$, where $i = 1, \cdots, n-1$.

Fig. 19 gives an illustration of the above divide-and-conquer strategy.

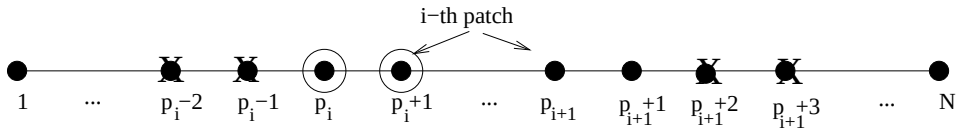


Figure 19: An illustration for the divide-and-conquer strategy. Circles indicate constraining points used in the Initialization step. Crosses are the constraining points that are used in step 2.

The above strategy is extremely useful when some computers do not have enough

memory (RAM) to run the optimization toolbox CVX with high-dimensional curves. In fact in Section 3.3, to analyze 4990-dimensional tonnage data, we applied the above divide-and-conquer strategy with $p_1 = 1663$, $p_2 = 3326$, and $p_3 = 4990$. The result is quite satisfactory: in nearly all cases, the algorithm converges right after the first iteration.

3.3 Numerical Experiments

We present a simulation study in Section 3.3.1. A real data case is presented in Section 3.3.2. In particular, IBC is compared with the fused lasso support vector classifier for the real data in Section 3.3.3.

3.3.1 Synthetic Data

We consider two baseline functions: $f_1(t) = \frac{\sin(t)}{t}$ and $f_2(t) = -\frac{\sin(t)}{t}$. They are drawn in Fig. 20; the red (dashed) and blue (solid) curves are $f_1(t)$ and $f_2(t)$, respectively. To discretize, we assume equally spaced points in the domain $[-4\pi, 4\pi]$. We have two

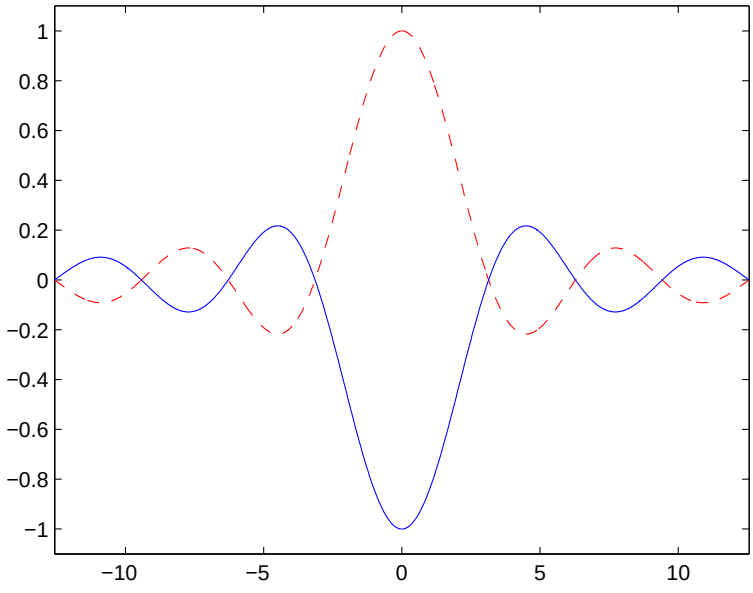


Figure 20: Two artificial baselines.

discrete baselines:

$$b_i(x) = f_i(-4\pi + \frac{x}{N} \cdot 8\pi), \quad x = 0, \cdots, N; i = 1, 2,$$

where N is the length of the curve. We generate the curves according to the following:

1. Generate random variables $\sigma(x) \overset{\text{i.i.d.}}{\sim} \text{Exponential}(\tau), x = 0, 1, \cdots, N$;
2. Generate the curve $c(x) \overset{\text{i.i.d.}}{\sim} \text{Laplace}(b(x), \sigma(x)), x = 0, 1, ..., N$.

According to the above procedure, we generate n curves (i.e., $c(\cdot)$'s) for each group. Fig. 21 illustrates the simulated data with $N = 155$, $\tau = 0.6$, $n = 10$. Note that

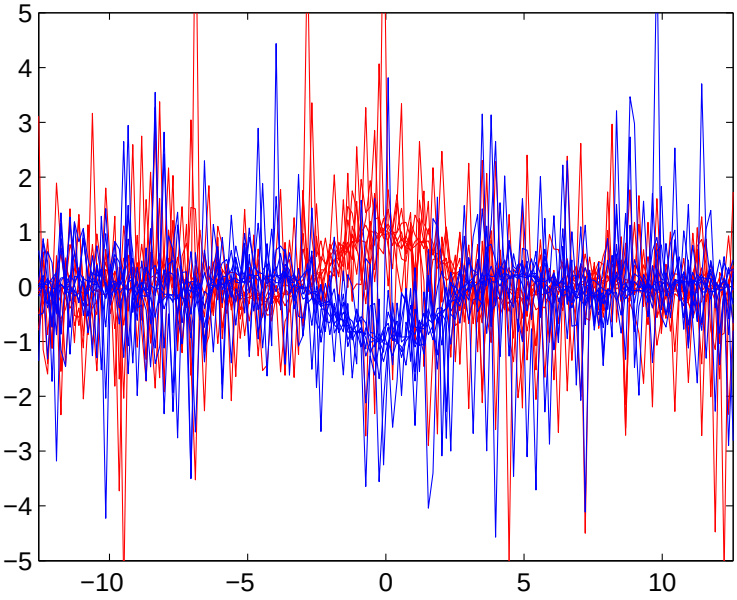
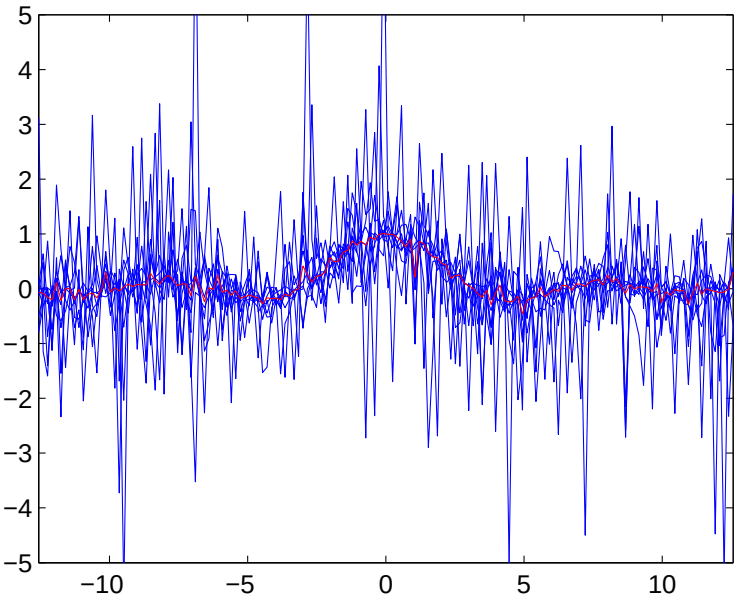


Figure 21: Simulated data with $N = 155$, $\tau = 0.6$, $n = 10$. The curves are very noisy.

the observed curves are extremely noisy. The proposed method is applied to this simulated data set. Fig. 22 presents the two estimated baselines. Fig. 23 shows the optimal value of objective function (i.e., $h(s)$) versus the length of interval (i.e., s), and the corresponding convex hull. Note that the number of admissible lengths is just 14 (out of 156). The full list of the admissible lengths and their corresponding

(a) Ten curves from class 1



(b) Ten curves from class 2

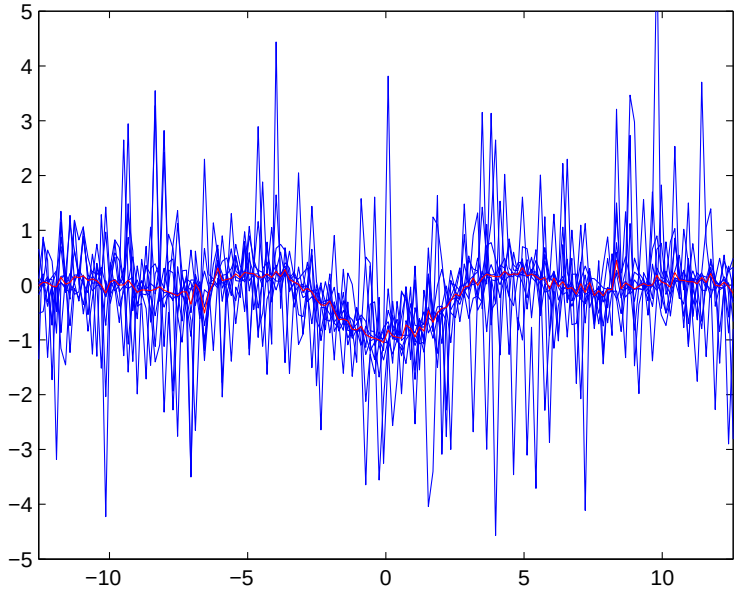
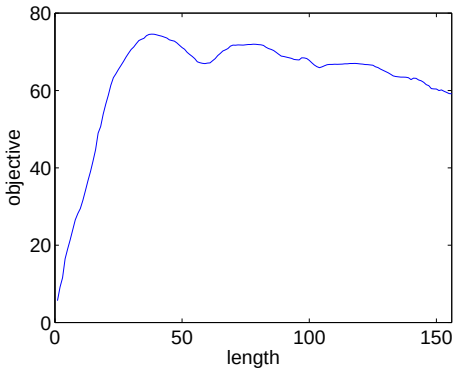


Figure 22: Baseline curves (red) for the two simulated groups. The curves are 156 dimensional. There are 10 curves in each group. In both classes, the estimated baselines are close to the true baselines.

(a) Maximum objective values



(b) The convex hull

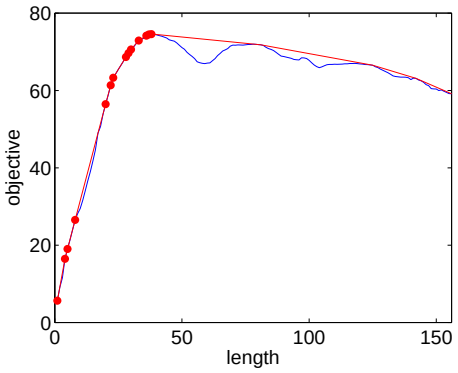


Figure 23: The entire objective function (a) and its convex hull (b). Note that the number of vertices on the convex hull is much smaller than the maximum length.

Table 5: Admissible lengths and the corresponding optimal intervals.

Length	Start	End	Length	Start	End
1	84 (0.8918)	84 (0.8918)	28	65 (−2.1890)	92 (2.1890)
4	81 (0.4054)	84 (0.8918)	29	64 (−2.3511)	92 (2.1890)
5	80 (0.2432)	84 (0.8918)	30	63 (−2.5133)	92 (2.1890)
8	80 (0.2432)	87 (1.3782)	33	63 (−2.5133)	95 (2.6754)
20	68 (−1.7025)	87 (1.3782)	36	60 (−2.9997)	95 (2.6754)
22	66 (−2.0268)	87 (1.3782)	37	60 (−2.9997)	96 (2.8376)
23	65 (−2.1890)	87 (1.3782)	38	60 (−2.9997)	97 (2.9997)

optimal intervals is provided in Table 5. Note that when the length is 38, the objective value is maximized, and the optimal interval is $(-2.9997, 2.9997)$. This is exactly the "true" optimal interval.

We repeat the entire procedures for 40 times to verify the consistency. Fig. 24 presents the optimal intervals for $n = 10, 30$, and 50; each horizontal line indicates the position of the computed optimal interval. Recall that visually, our simulated data (i.e., Fig. 21) can hardly be distinguished. Fig. 24 shows that if n is large enough, we can consistently find the optimal interval on which the curves are well separable. This demonstrates the consistency of our estimator. Fig. 25 shows the separation

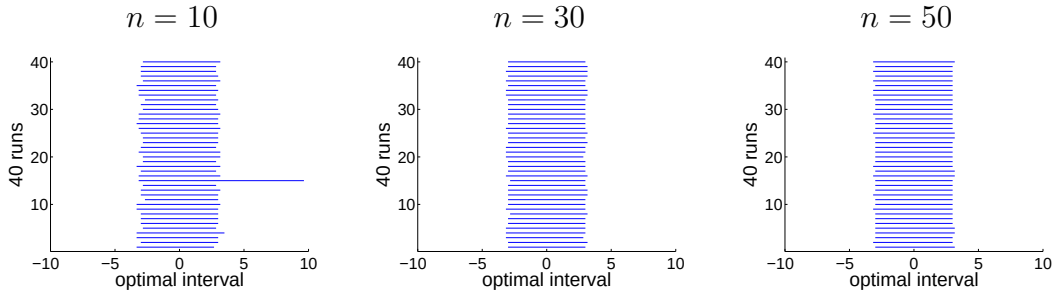


Figure 24: Optimal intervals in 40 experiments with $\tau = 0.6$ and various n's. In most cases, the estimated optimal intervals are close to the true optimal intervals.

of the two classes of simulated curves within the computed optimal interval. We observe that within the identified optimal interval, the two classes of curves are well separated.

To check whether our method works well for worse case (i.e., noisier case), we increase the value of τ from 0.6 to 1. Fig. 26 shows that it is more difficult to distinguish the curves than the previous case (i.e, $\tau = 0.6$). The same experiment is performed. Fig. 27 shows that as n increases, our method becomes more consistent in locating the optimal interval for much noisier curves.

It will be also interesting to see the results for the case when N is large. For this purpose, we perform the same experiment for $N = 305, 605$, and 1205. We set $\tau = 1$ (i.e., very noisy case). The results on the optimal intervals are in Fig. 28.

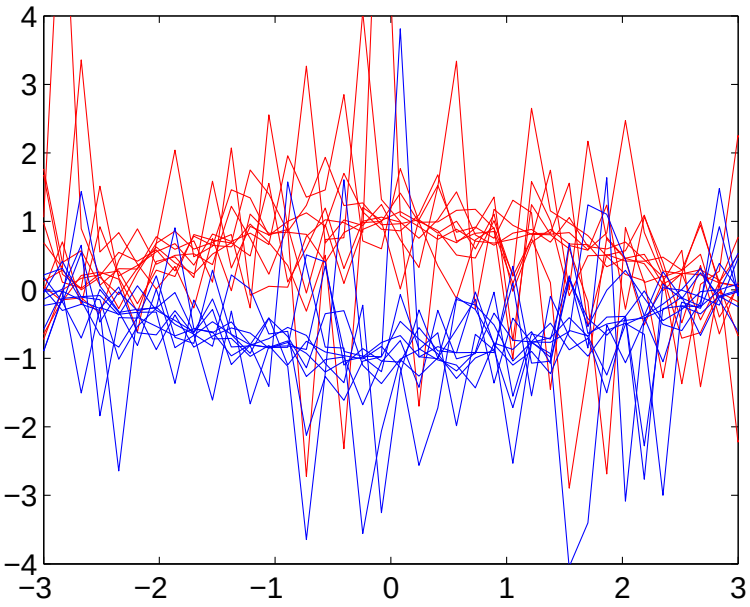


Figure 25: Segments of curves within the computed optimal interval. We observe that the two classes of curves are clearly separated.

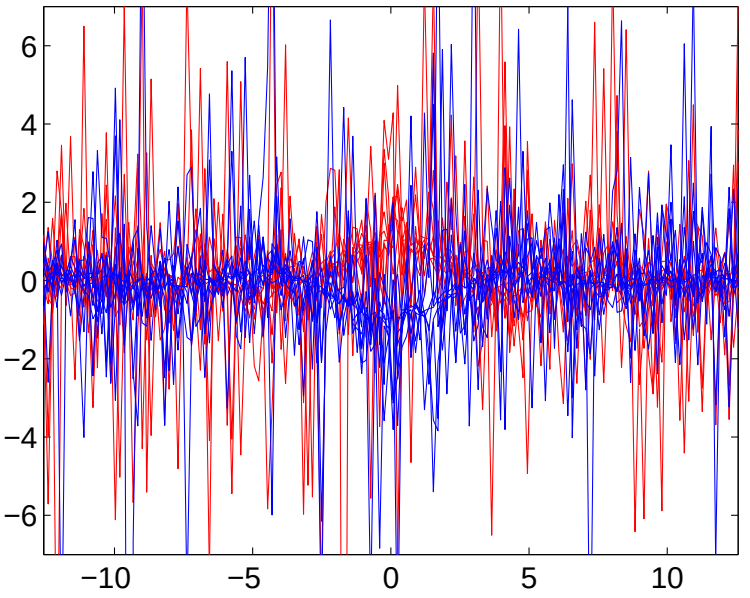


Figure 26: Simulated data with $\tau = 1$, $n = 10$. This case is more difficult to classify than the previous one.

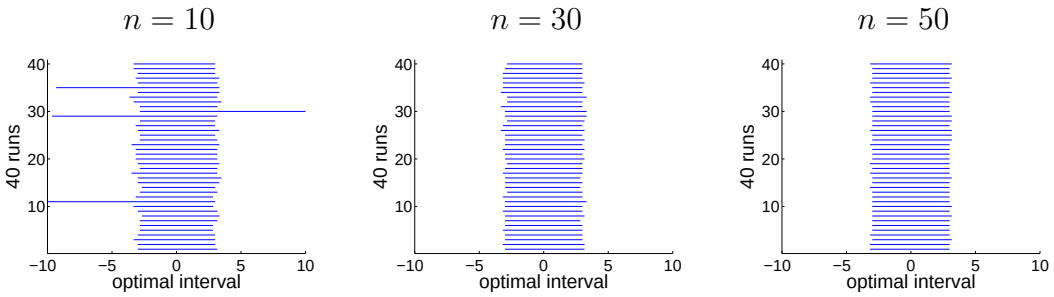


Figure 27: Optimal intervals with $\tau = 1$ and various n's. Again the estimated optimal intervals are close to the true optimal interval.

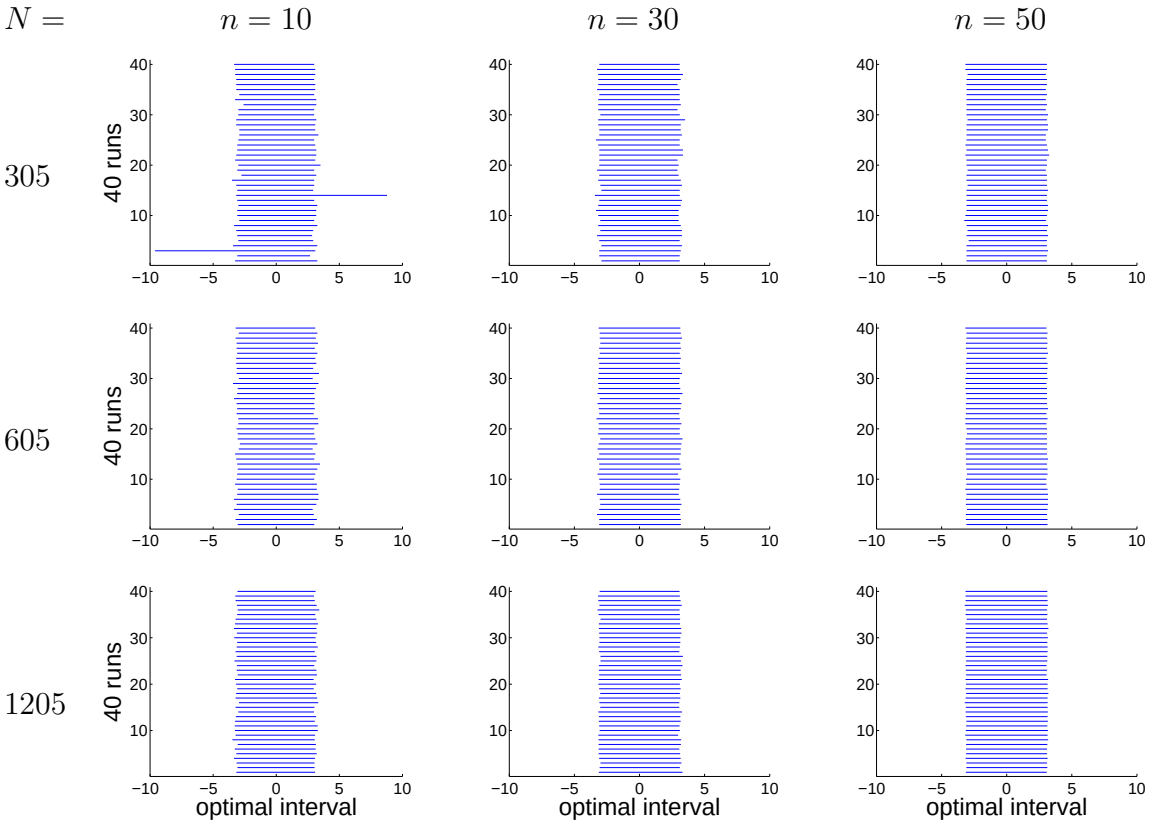


Figure 28: Optimal intervals with $\tau = 1$ and various n's and N's.

Table 6 shows the execution time for our procedures. These procedures include all the steps, including finding baselines, computing the value of objective function, identifying an optimal interval. For 40 runs, we measure the execution time for each run, and then take the average. The execution time is quite impressive, considering the dimension of the curves and the numbers of samples.

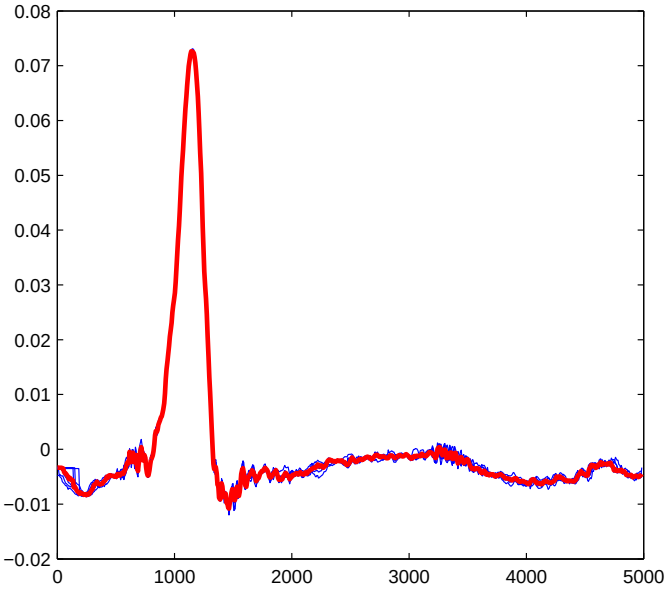
Table 6: Execution time (in seconds).

	n=10	30	50
N=75	0.8	2.1	4.2
155	1.3	4.3	9.0
305	2.4	9.1	20.4
605	4.6	16.6	39.9
1205	9.0	38.9	81.7

3.3.2 A Real Case: Tonnage Data from a Stamping Process

The developed IBC has been tested with tonnage curves obtained from a real production environment. Two sets of curves under normal and faulty conditions are used. The stamped parts are manufactured from a progressive die. The total tonnage curve is measured as the summation of all forming operations within the press. The faulty condition is that one part is missing in one of the die stations, which introduces some tonnage curve change in one of the segments. However, which segments (i.e., intervals) of the tonnage curve have the most sensitive difference is unknown. Such intervals are identified using the IBC. Fig. 29 presents the baselines (in red) for the two classes. We highly recommend the readers to view this figure in color. It is more interesting to compare the two estimated baselines. Fig. 30 presents a comparison. For the tonnage data, Fig. 31 (a) shows the optimized objective value and the corresponding interval length, and Fig. 31 (b) shows the convex hull. MATLAB offers built-in function "`convhull`" for calculating convex hull for a curve given by

(a) Faulty



(b) Normal

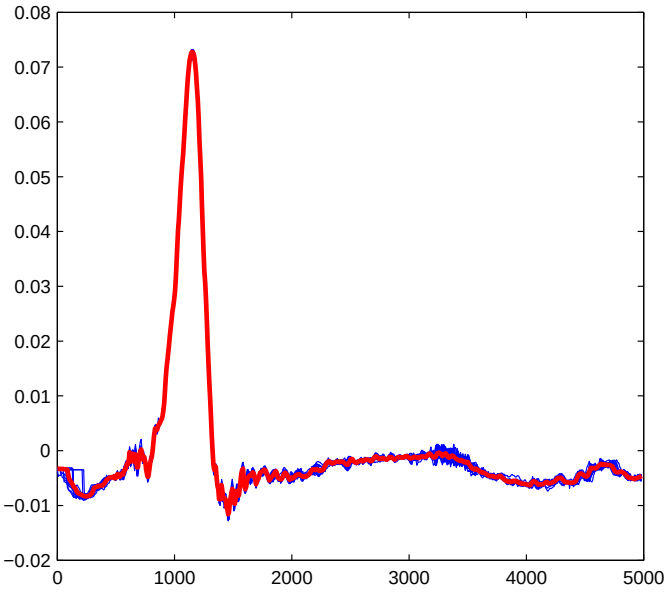


Figure 29: The computed baselines for faulty (a) and normal (b) tonnage curves respectively.

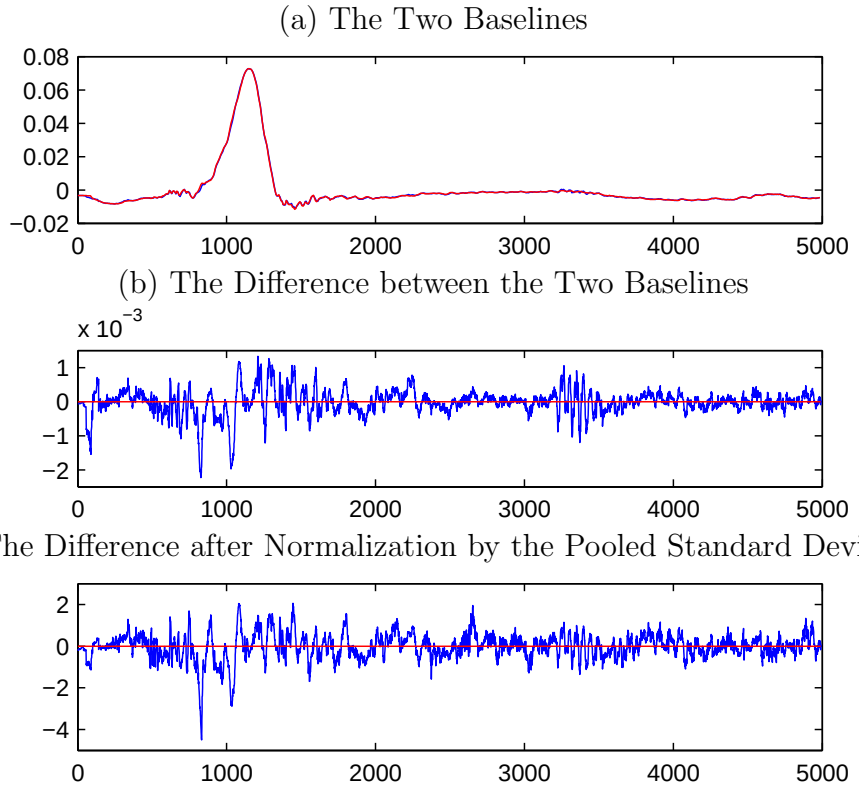


Figure 30: A comparison of the two estimated baselines.

a set of points. Note that the number of admissible lengths is only 49, considering the number of possible lengths is 4990. A full list of admissible lengths and their corresponding optimal intervals is in Table 7. Note that if we are required to use intervals shorter than 35, then the optimal one is $[808, 841]$ (with length 34; 35 is not admissible), which is very close to the interval $[810, 845]$ that is a known optimal interval for separation based on physical knowledge. In this case, the output of IBC is consistent with the expert's knowledge.

To show the effectiveness of IBC, we compare the original curves and the segmented curves within the optimal interval in Fig. 16. Fig. 16 (a) plots all the curves ($N = 4990$ dimensional) that are faulty (in red) and normal (in blue), respectively. Visually, the two classes are indistinguishable. However, in Fig. 16 (b), the two classes have been observed to be well separated in interval $(808, 841)$.

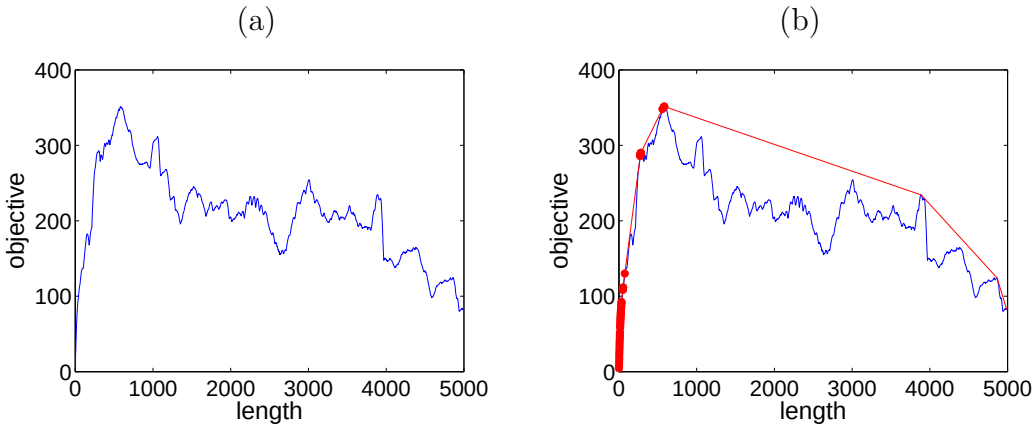


Figure 31: (a) Maximum objective values; (b) Convex hull.

Table 7: Admissible lengths and the corresponding optimal intervals.

Length	Start	End	Length	Start	End	Length	Start	End
1	832	832	19	818	836	53	806	858
2	831	832	20	818	837	54	805	858
3	830	832	21	817	837	76	783	858
4	829	832	22	816	837	274	783	1056
5	828	832	23	815	837	275	782	1056
6	828	833	24	815	838	276	782	1057
7	827	833	25	814	838	278	780	1057
8	826	833	26	813	838	279	779	1057
9	826	834	28	811	838	280	779	1058
10	825	834	29	811	839	282	777	1058
11	824	834	30	811	840	283	777	1059
12	824	835	32	809	840	560	500	1059
13	823	835	33	808	840	581	479	1059
14	822	835	34	808	841	582	478	1059
16	820	835	50	808	857	585	478	1062
17	819	835	51	808	858			
18	818	835	52	807	858			

3.3.3 Comparison as a Classifier

Using the identified optimal interval from the previous subsection and the classification rule as in Section 3.2.5, the tonnage curves are classified into two classes. The performance of IBC is verified in two ways: (a) We show that much lower misclassification error rate can be achieved by using data within the optimal interval rather than the entire data. (b) IBC produces similar solutions to those from the fused lasso support vector classifier, but is more stable, and guarantees the unique optimal interval.

We first examine the approach of using the optimal 'segment' of data. If we completely ignore the optimal interval, and carry out a classification on the entire domain, using the classification rule as in Section 3.2.5, the error rate can be much higher than doing so by combining with the optimal interval. In our experiment of 7-fold cross validation, all tonnage curves are divided into 7 groups. For each group, we use remaining 6 groups as training data, and the left-out group as testing data. In this scenario, 18 (out of 73) times the classification result is erroneous. Now if we insert the optimal interval selection immediately after the step of finding the baselines (i.e., before applying the classification rule), as shown in Table 8, the number of misclassified curves is much smaller. This experiment demonstrates that (at least in this setting), adding a step of selecting optimal interval is advantageous.

Secondly, IBC is compared with the fused lasso support vector classifier (SVC) that also produces interval-like solutions. For IBC, 7-fold cross-validation is repeated for every admissible length (i.e., 49 times) so that we can choose the optimal interval that minimizes the number of misclassified curves. For the fused lasso SVC, the values of the two tuning parameters $C1$ and $C2$ (upper bounds for the L_1-norms of the coefficients and their successive differences, respectively) should be set in advance. To search the optimal values of $C1$ and $C2$, 7-fold cross-validation is repeated 100 times, as both $C1$ and $C2$ are varied from 100 to 1000 with increment 100. The

Table 8: Number of misclassified curves according to the various admissible lengths in IBC.

Length	Number of errors	Length	Number of errors	Length	Number of errors
1	0	19	1	53	4
2	1	20	1	54	4
3	1	21	1	76	2
4	0	22	1	274	7
5	1	23	1	275	7
6	1	24	2	276	7
7	1	25	1	278	7
8	1	26	1	279	7
9	1	28	2	280	7
10	1	29	2	282	7
11	1	30	2	283	8
12	1	32	2	560	11
13	1	33	2	581	11
14	1	34	2	582	11
16	1	50	3	585	11
17	1	51	3		
18	1	52	4		

values minimizing the number of misclassified curves are chosen as the optimal values of $C1$ and $C2$. Table 8 and Table 9 show the results for IBC and the fused lasso SVC, respectively. We have two observations. One is that the results from IBC are very stable with various lengths, while those from the fused lasso SVC is rather bumpy. Indeed, when using IBC, among the 49 admissible lengths, 24 (roughly 50 percent) admissible lengths produce either 0 or 1 error. Moreover, most admissible lengths (37 out of 49) produce less than 4 errors. In contrast, the performance of the fused lasso SVC substantially depends on the values of $C1$ and $C2$, and therefore the choices of the two values are very critical. This implies bumpy response surface that is associated with the response in Table 9. Hence, finding optimal $(C1, C2)$ is not

going to be easy.

Another observation is that while IBC produces the unique optimal interval given a length, the fused lasso SVC produces multiple intervals. In fact, with the optimal choices of $C1 = 1000$ and $C2 = 200$, the fused lasso SVC estimates the nonzero coefficients as: $\hat{\beta}_{824} = \cdots = \hat{\beta}_{833} = 50.79$, $\hat{\beta}_{834} = 1.05$, $\hat{\beta}_{1029} = \cdots \hat{\beta}_{1035} = 14.99$, $\hat{\beta}_{1082} = \cdots = \hat{\beta}_{1092} = -34.21$, $\hat{\beta}_{1093} = -9.71$. Note that the first interval (with the largest coefficients) is the same as the optimal interval from IBC with length 11, which suggests that (1) an underlying optimal interval does exist for this data set; (2) both methods (IBC & fused lasso SVC) recover such an interval in some sense. However, IBC explicitly searches for such an interval, while the fused lasso SVC does it without a guarantee. In some engineering problems, e.g., analyzing the stamping process, it is more convenient to adopt IBC instead of SVC.

Table 9: Number of misclassified curves according to the various $C1$ and $C2$ in the fused SVC. Note that the misclassification rate fluctuates, indicating difficulty in searching for a minimum.

$C1 \backslash C2$	100	200	300	400	500	600	700	800	900	1000
100	34	37	46	16	34	7	7	7	7	7
200	34	34	43	46	37	19	55	46	16	7
300	28	25	7	7	52	46	28	46	28	28
400	16	25	16	16	16	34	25	46	37	37
500	7	25	16	16	7	34	16	34	16	34
600	16	16	16	16	19	16	7	16	34	34
700	6	24	7	16	16	7	16	7	7	7
800	4	4	24	23	4	4	4	4	4	4
900	3	3	12	3	2	3	35	13	3	3
1000	3	0	1	1	11	1	1	2	2	2

3.4 Conclusion

We formulate a special class of classification problems (from data mining) into an optimization problem, and solve the problem with analytical study and demonstrate with real data. In particular, we propose an interval based classifier. Its application

is demonstrated by classifying some tonnage data. Fast numerical algorithms are derived. The proposed method is efficient—its order of complexity is nearly linear to the length of the curve. In applications of IBC, we demonstrate its speed and ability to deal with high-dimensional data. In many applications, problems that require interval based classifier may benefit from using IBC. Our computational method has a root in operations research. IBC should have many applications in data mining, when the features are based on single intervals. Our future research includes multiple (more than 2) classes scenario. In this case, a simple approach for identifying an optimal interval is to maximize the minimum of the pairwise distances of multiple baselines. However, if the number of class is large, this approach with a single optimal interval may not be effective; adopting multiple optimal intervals can be considered. The case of multiple optimal intervals and integration with other signal processing methods, e.g., wavelet transforms will be interesting future work.

APPENDIX A

PROOFS

A.1 Derivation of (2.12)

We can easily verify the following with standard calculation:

$$\int_{x_i}^{x_{i+1}} \{f^{(2)}(x)\}^2 dx = \frac{h}{3}\delta_i^2 + \frac{h}{3}\delta_{i+1}^2 + \frac{h}{3}\delta_i\delta_{i+1}, \quad i = 1, \cdots, n-1, \tag{1.53}$$

where $\delta_i = f^{(2)}(x_i), i = 2, \cdots, n-1$, $\delta_1 = \delta_n = 0$, and $h = x_{i+1} - x_i, 1 \le i \le n-1$. With the assumption that $\lambda(x) \equiv \lambda_i$ for $x \in [x_i, x_{i+1}], 1 \le i \le n-1$, using (1.53), we have

$$\begin{aligned} J^\star(\boldsymbol{\lambda}; \mathbf{f}) &= (\mathbf{y}-\mathbf{f})^T(\mathbf{y}-\mathbf{f}) + \int_0^1 \lambda(x)\{f^{(2)}(x)\}^2 dx \\ &= (\mathbf{y}-\mathbf{f})^T(\mathbf{y}-\mathbf{f}) + \frac{h}{3}\sum_{i=1}^{n-1} \lambda_i(\delta_i^2 + \delta_{i+1}^2 + \delta_i\delta_{i+1}) \\ &= (\mathbf{y}-\mathbf{f})^T(\mathbf{y}-\mathbf{f}) + \frac{h}{3}\delta^T\mathbf{M}(\boldsymbol{\lambda})\delta. \end{aligned}$$

A.2 Proof of Theorem 1.2.5

To prove Theorem 1.2.5, we extend the proof of Theorem 1 in [30]. Note that $\boldsymbol{\lambda}$ can be factorized as $\lambda_1(1, \lambda_2/\lambda_1, \cdots, \lambda_{n-1}/\lambda_1)$ where λ_1 is the first element of $\boldsymbol{\lambda}$, and let $\tilde{\boldsymbol{\lambda}}$ denote $(1, \lambda_2/\lambda_1, \cdots, \lambda_{n-1}/\lambda_1)$. Recall $\mathbf{S}_n(\boldsymbol{\lambda}) = (\mathbf{I} + \boldsymbol{\Sigma}_n(\boldsymbol{\lambda}))^{-1}$ where $\boldsymbol{\Sigma}_n(\boldsymbol{\lambda}) = \frac{h}{3}\mathbf{Q}^T\mathbf{M}^{-1}\mathbf{M}(\boldsymbol{\lambda})\mathbf{M}^{-1}\mathbf{Q}$, and note that $\boldsymbol{\Sigma}_n(\boldsymbol{\lambda}) = \lambda_1\boldsymbol{\Sigma}_n(\tilde{\boldsymbol{\lambda}})$. Let $0 \le \tilde{\tau}_1 \le \tilde{\tau}_2 \le \cdots \le \tilde{\tau}_n$ be the eigenvalues of $\boldsymbol{\Sigma}_n(\tilde{\boldsymbol{\lambda}})$. As in [30], the key to prove the above theorem is to establish the following three inequalities: There exist $\delta_1, \delta_2 > 0$ and $a_n \to 0$ such that

$$P\left\{\inf_{\lambda_1 \ge 0}\inf_{\tilde{\boldsymbol{\lambda}} \ge 0} \lambda_1^2 \sum_{i=1}^n (f(x_i) + \epsilon_i)^2(\lambda_1 + \tilde{\tau}_i)^{-2}/Q_n(\lambda_1) \le a_n\right\} \le \delta_2/2,$$

$$P\left\{\sup_{\lambda_1 \ge 0}\sup_{\tilde{\boldsymbol{\lambda}} \ge 0} \frac{\lambda_1^2}{nQ_n(\lambda_1)} \sum_{i=1}^n (\lambda_1 + \tilde{\tau}_i)^{-1}\left|\sum_{i=1}^n \frac{\epsilon_i^2 - \sigma^2}{\lambda_1 + \tilde{\tau}_i}\right| \ge a_n\delta_1\right\} \le \delta_2/2,$$

$$P\left\{\sup_{\lambda_1\geq 0}\sup_{\tilde{\boldsymbol{\lambda}}\geq 0}\frac{\lambda_1^2}{nQ_n(\lambda_1)}\sum_{i=1}^{n}(\lambda_1+\tilde{\tau}_i)^{-1}\left|\sum_{i=1}^{n}\frac{f(x_i)\epsilon_i}{\lambda_1+\tilde{\tau}_i}\right|\geq a_n\delta_1\right\}\leq \delta_2/2,$$

where $Q_n(\lambda_1)=\lambda_1^2\sum_{i=1}^{n}(\lambda_1+\tilde{\tau}_i)^{-2}(f(x_i)^2+\sigma^2)$. A careful scrutiny at the proof in [30] revealed that the choice of δ_1, δ_2, and a_n does not depend on the values of $\tilde{\tau}_i$, so the argument in [30] is sufficient to establish Theorem 1.2.5. To avoid repeating a lengthy discussion, we omit a detailed proof here.

A.3 Proof of Theorem 1.2.6

We first prove (2.18). Let $\mathbf{A}_n(\hat{\boldsymbol{\lambda}})=\mathbf{I}_n-\mathbf{S}_n(\hat{\boldsymbol{\lambda}})$. Rewrite (2.18) as

$$\begin{aligned}
&2\left|\frac{\sigma^2\mathrm{tr}\mathbf{A}_n(\hat{\boldsymbol{\lambda}})}{n\|\mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{y}_n\|^2}\langle\boldsymbol{\epsilon}_n,\mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{y}_n\rangle-\frac{\sigma^4(\mathrm{tr}\mathbf{A}_n(\hat{\boldsymbol{\lambda}}))^2}{n\|\mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{y}_n\|^2}-n^1\|\boldsymbol{\epsilon}_n\|^2+\sigma^2\right|/L_n(\hat{\boldsymbol{\lambda}})\\
\leq\ & 2\sigma^2\mathrm{tr}\mathbf{A}_n(\hat{\boldsymbol{\lambda}})\left|\langle\boldsymbol{\epsilon}_n,\mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{f}_n\rangle\right|/n\|\mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{y}_n\|^2L_n(\hat{\boldsymbol{\lambda}}) &(3.54)\\
&+2\sigma^2\mathrm{tr}\mathbf{A}_n(\hat{\boldsymbol{\lambda}})\left|\langle\boldsymbol{\epsilon}_n,\mathbf{S}_n(\hat{\boldsymbol{\lambda}})\boldsymbol{\epsilon}_n\rangle-\sigma^2\mathrm{tr}\mathbf{S}_n(\hat{\boldsymbol{\lambda}})\right|/n\|\mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{y}_n\|^2L_n(\hat{\boldsymbol{\lambda}}) &(3.55)\\
&+2\left|\left(\frac{\sigma^2\mathrm{tr}\mathbf{A}_n(\hat{\boldsymbol{\lambda}})}{\|\mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{y}_n\|^2}-1\right)(\sigma^2-n^{-1}\|\boldsymbol{\epsilon}_n\|^2)\right|/L_n(\hat{\boldsymbol{\lambda}}). &(3.56)
\end{aligned}$$

It suffices to show (3.54), (3.55), (3.56) tend to 0. Note that (2.16) is equivalent to $n^{-1}\|\mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{y}_n\|^2\to\sigma^2$. For (3.54) and (3.55), using (2.16), it suffices to show

$$\sup_{\boldsymbol{\lambda}>0}\left|n^{-1}\langle\boldsymbol{\epsilon}_n,\mathbf{A}_n(\boldsymbol{\lambda})\mathbf{f}_n\rangle\right|/\mathbb{E}L_n(\boldsymbol{\lambda})\to 0, \tag{3.57}$$

$$\sup_{\boldsymbol{\lambda}>0}n^{-1}\left|\sigma^2\mathrm{tr}\mathbf{S}_n(\boldsymbol{\lambda})-\langle\boldsymbol{\epsilon}_n,\mathbf{S}_n(\boldsymbol{\lambda})\boldsymbol{\epsilon}_n\rangle\right|/\mathbb{E}L_n(\boldsymbol{\lambda})\to 0, \tag{3.58}$$

and

$$\sup_{\boldsymbol{\lambda}>0}|L_n(\boldsymbol{\lambda})/\mathbb{E}L_n(\boldsymbol{\lambda})-1|\to 0. \tag{3.59}$$

For (3.57), given any $\delta>0$, we have by Chebychev inequality that

$$P\{\sup_{\boldsymbol{\lambda}>0}\left|n^{-1}\langle\boldsymbol{\epsilon}_n,\mathbf{A}_n(\boldsymbol{\lambda})\mathbf{f}_n\rangle\right|/\mathbb{E}L_n(\boldsymbol{\lambda})>\delta\}\leq\delta^{-2}n^{-2}\mathbb{E}\|\boldsymbol{\epsilon}_n\|^2\|\mathbf{A}_n(\boldsymbol{\lambda})\mathbf{f}_n\|^2(\mathbb{E}L_n(\boldsymbol{\lambda}))^{-2}. \tag{3.60}$$

Using the fact that $n^{-1}\|\mathbf{A}_n(\boldsymbol{\lambda})\mathbf{f}_n\|^2\leq\mathbb{E}L_n(\boldsymbol{\lambda})$ and **(A.1)**, (3.60) goes to 0. For (3.58), by using $\mathbb{E}\left(\sigma^2\mathrm{tr}\mathbf{S}_n(\boldsymbol{\lambda})-\langle\boldsymbol{\epsilon}_n,\mathbf{S}_n(\boldsymbol{\lambda})\boldsymbol{\epsilon}_n\rangle\right)^2\leq 2\sigma^4\mathrm{tr}\mathbf{S}_n^2(\boldsymbol{\lambda})$, and $\sigma^2n^{-1}\mathrm{tr}\mathbf{S}_n^2(\boldsymbol{\lambda})\leq\mathbb{E}L_n(\boldsymbol{\lambda})$,

it can be proved in a similar way to prove (3.57). For (3.59), it suffices to show the following

$$\sup_{\boldsymbol{\lambda}>0} n^{-1} \left|\langle \mathbf{A}_n(\boldsymbol{\lambda})\mathbf{f}_n, \mathbf{S}_n(\boldsymbol{\lambda})\boldsymbol{\epsilon}_n\rangle\right| / \mathbb{E}L_n(\boldsymbol{\lambda}) \to 0, \tag{3.61}$$

and

$$\sup_{\boldsymbol{\lambda}>0} n^{-1} \left|\|\mathbf{S}_n(\boldsymbol{\lambda})\boldsymbol{\epsilon}_n\|^2 - \sigma^2 \mathrm{tr}\mathbf{S}_n^2(\boldsymbol{\lambda})\right| / \mathbb{E}L_n(\boldsymbol{\lambda}) \to 0. \tag{3.62}$$

Notice that (3.61) and (3.62) can be proved in the same way as (3.57) and (3.58), respectively. Now only (3.56) remains to be proved. To prove (3.56), it suffices to show

$$\left|\sigma^2 n^{-1}\mathrm{tr}\mathbf{A}_n(\hat{\boldsymbol{\lambda}}) - n^{-1}\|\mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{y}_n\|^2\right| \left|\sigma^2 - n^{-1}\|\boldsymbol{\epsilon}_n\|^2\right| / L_n(\hat{\boldsymbol{\lambda}}) \to 0.$$

Note that since $\|\mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{y}_n\|^2 = \|\boldsymbol{\epsilon}_n\|^2 + 2\langle \boldsymbol{\epsilon}_n, \mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{f}_n\rangle - 2\langle \boldsymbol{\epsilon}_n, \mathbf{S}_n(\hat{\boldsymbol{\lambda}})\boldsymbol{\epsilon}_n\rangle + \|\mathbf{f}_n - \hat{\mathbf{f}}_n\|^2$, we have

$$\begin{aligned} & \left|\sigma^2 n^{-1}\mathrm{tr}\mathbf{A}_n(\hat{\boldsymbol{\lambda}}) - n^{-1}\|\mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{y}_n\|^2\right| \\ \leq & \left|\sigma^2 - \frac{1}{n}\|\boldsymbol{\epsilon}_n\|^2\right| + L_n(\hat{\boldsymbol{\lambda}}) + \frac{2}{n}\left|\langle \boldsymbol{\epsilon}_n, \mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{f}_n\rangle\right| + \frac{2}{n}\left|\langle \boldsymbol{\epsilon}_n, \mathbf{S}_n(\hat{\boldsymbol{\lambda}})\boldsymbol{\epsilon}_n\rangle - \sigma^2\mathrm{tr}\mathbf{S}_n(\hat{\boldsymbol{\lambda}})\right| \\ & + n^{-1}\sigma^2\mathrm{tr}\mathbf{S}_n(\hat{\boldsymbol{\lambda}}). \end{aligned}$$

Then using (3.57), (3.58), and (2.16), it suffices to show

$$(\sigma^2 - n^{-1}\|\boldsymbol{\epsilon}_n\|^2)^2 / L_n(\hat{\boldsymbol{\lambda}}) \to 0, \tag{3.63}$$

and

$$(n^{-1}\mathrm{tr}\mathbf{S}_n(\hat{\boldsymbol{\lambda}})) \left|\sigma^2 - n^{-1}\|\boldsymbol{\epsilon}_n\|^2\right| / L_n(\hat{\boldsymbol{\lambda}}) \to 0. \tag{3.64}$$

By **(A.1)**, (3.59), and the central limit theorem, we have (3.63). Using (3.63), (3.59), **(A.1)**, and the fact that $(n^{-1}\mathrm{tr}\mathbf{S}_n(\hat{\boldsymbol{\lambda}}))^2 \leq \mathbb{E}L_n(\hat{\boldsymbol{\lambda}})$, we have (3.64). Hence, we complete the proof of (2.18). Now it remains to prove (2.19). The numerator of (2.19) can be rewritten as

$$\begin{aligned} & n^{-1}\|\tilde{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}}) - \hat{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}})\|^2 = \left(\frac{n^{-1}\sigma^2\mathrm{tr}\mathbf{A}_n(\hat{\boldsymbol{\lambda}})}{n^{-1}\|\mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{y}_n\|^2} - 1\right)^2 n^{-1}\|\mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{y}_n\|^2 \\ = & \frac{\left[(\sigma^2 - \frac{1}{n}\|\boldsymbol{\epsilon}_n\|^2) - L_n(\hat{\boldsymbol{\lambda}}) - \frac{2}{n}\langle \boldsymbol{\epsilon}_n, \mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{f}_n\rangle + \frac{2}{n}\langle \boldsymbol{\epsilon}_n, \mathbf{S}_n(\hat{\boldsymbol{\lambda}})\boldsymbol{\epsilon}_n\rangle - \frac{1}{n}\sigma^2\mathrm{tr}\mathbf{S}_n(\hat{\boldsymbol{\lambda}})\right]^2}{n^{-1}\|\mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{y}_n\|^2}. \end{aligned}$$

Since $n^{-1}\|\mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{y}_n\|^2 \to \sigma^2$, to prove (2.19), it suffices to show the following:

$$(\sigma^2 - n^{-1}\|\boldsymbol{\epsilon}_n\|^2)^2/L_n(\hat{\boldsymbol{\lambda}}) \to 0, \tag{3.65}$$

$$\left(n^{-1}\langle\boldsymbol{\epsilon}_n, \mathbf{A}_n(\hat{\boldsymbol{\lambda}})\mathbf{f}_n\rangle\right)^2/L_n(\hat{\boldsymbol{\lambda}}) \to 0, \tag{3.66}$$

$$(n^{-1}\mathrm{tr}\mathbf{S}_n(\hat{\boldsymbol{\lambda}}))^2/L_n(\hat{\boldsymbol{\lambda}}) \to 0, \tag{3.67}$$

and

$$\left(n^{-1}\langle\boldsymbol{\epsilon}_n, \mathbf{S}_n(\hat{\boldsymbol{\lambda}})\boldsymbol{\epsilon}_n\rangle\right)^2/L_n(\hat{\boldsymbol{\lambda}}) \to 0. \tag{3.68}$$

Notice that (3.65) is the same as (3.63). Using (3.57), (3.66) can be easily proven. (3.67) follows from (2.17) and (3.59). Finally, (3.68) follows from (3.58) and (3.67).

A.4 Proof of Theorem 1.2.7

We start with some basic linear algebra facts. It can be shown that for matrix $\Phi = \mathbf{M}^{-1}\mathbf{Q}$, we have $\Phi_{(1,2,\cdots,n-2)\times(2,3,\cdots,n-1)} = 2\mathbf{I} - 6\mathbf{M}^{-1}$. The above can be verified by multiplying $\mathbf{M}$ on both sides; it also indicates that we only need to consider the eigenvalues of the matrix $(2\mathbf{I}-6\mathbf{M}^{-1})^T\mathbf{M}(\boldsymbol{\lambda})(2\mathbf{I}-6\mathbf{M}^{-1})$, because one can then apply the interlacing theorem to describe the eigenvalues of $\boldsymbol{\Sigma}(\boldsymbol{\lambda})$. Formally, we have

$$(2\mathbf{I} - 6\mathbf{M}^{-1})^T\mathbf{M}(\boldsymbol{\lambda})(2\mathbf{I} - 6\mathbf{M}^{-1}) = \boldsymbol{\Sigma}(\boldsymbol{\lambda})_{(2,\cdots,n-1)\times(2,\cdots,n-1)}. \tag{4.69}$$

That is, the left side is a principal submatrix of $\boldsymbol{\Sigma}(\boldsymbol{\lambda})$. It is known [9] that $\mathbf{M}^{-1} = \boldsymbol{\Gamma}^T\mathbf{D}_1\boldsymbol{\Gamma}$, where $\mathbf{D}_1 = \mathrm{diag}\left\{1/\left(2+\cos\frac{i\pi}{n}\right)\right\}, i = 1, 2, \cdots, n-2$, and $\Gamma_{jk} = \sqrt{\frac{2}{n}}\sin\frac{jk\pi}{n}, 1 \leq j, k \leq n-2$. Noticing that $\boldsymbol{\Gamma}$ is orthogonal, we can easily derive the following: $2\mathbf{I} - 6\mathbf{M}^{-1} = \boldsymbol{\Gamma}^T\ \mathbf{D}_2\boldsymbol{\Gamma}$, where $\mathbf{D}_2 = \mathrm{diag}\left\{\frac{2\cos\frac{i\pi}{n}-2}{2+\cos\frac{i\pi}{n}}\right\}, 1 \leq i \leq n-2$. Bringing this into (4.69), we have $(2\mathbf{I} - 6\mathbf{M}^{-1})^T\mathbf{M}(\boldsymbol{\lambda})(2\mathbf{I} - 6\mathbf{M}^{-1}) = \boldsymbol{\Gamma}^T\mathbf{D}_2\boldsymbol{\Gamma}\mathbf{M}(\boldsymbol{\lambda})\boldsymbol{\Gamma}^T\mathbf{D}_2\boldsymbol{\Gamma}$, which has the same eigenvalues as $\mathbf{D}_2\boldsymbol{\Gamma}\mathbf{M}(\boldsymbol{\lambda})\boldsymbol{\Gamma}^T\mathbf{D}_2$.

We need the following lemma.

Lemma A.4.1 *Suppose matrix* $\mathbf{A} \in \mathbb{R}^{n\times n}$ *and* $\mathbf{A}$ *is symmetric positive definite. Suppose the eigenvalues of* $\mathbf{A}$ *are* $\gamma_1(\mathbf{A}) \leq \gamma_2(\mathbf{A}) \leq \cdots \leq \gamma_n(\mathbf{A})$. *Let* $\mathbf{D}$ *be an diagonal*

matrix $\mathbf{D} = diag\{d_1, d_2, \cdots, d_n\}$, *where* $0 \leq d_1 \leq d_2 \leq \cdots d_n$. *Let* $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_n$ *be the eigenvalues of matrix* $\mathbf{DAD}$. *We have* $\gamma_1(\mathbf{A})d_i^2 \leq \tau_i \leq \gamma_n(\mathbf{A})d_i^2$.

Proof. According to the minimax theorem, we have

$$\tau_i = \sup_{\Omega_i} \inf_{x \in \Omega_i} \frac{x^T \mathbf{DAD} x^T}{x^T x},$$

where Ω_i is an i-dimensional linear subspace of $\mathbb{R}^n (\Omega_i \subset \mathbb{R}^n)$ and x is an n-dimensional vector. We can easily establish the following:

$$\gamma_1(\mathbf{A}) x^T \mathbf{D}^2 x \leq x^T \mathbf{DAD} x \leq \gamma_n(\mathbf{A}) x^T \mathbf{D}^2 x.$$

Applying operator $\sup_{\Omega_i} \inf_{x \in \Omega_i}$ on all the three terms above, we have

$$\gamma_1(\mathbf{A}) \sup_{\Omega_i} \inf_{x \in \Omega_i} \frac{x^T \mathbf{D}^2 x}{x^T x} \leq \tau_i \leq \gamma_n(\mathbf{A}) \sup_{\Omega_i} \inf_{x \in \Omega_i} \frac{x^T \mathbf{D}^2 x}{x^T x}.$$

Notice that the first term above is $\gamma_1(\mathbf{A})d_i^2$ and the last term is $\gamma_n(\mathbf{A})d_i^2$. □

Due to the above lemma, if $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_{n-2}$ are the eigenvalues of $\mathbf{D}_2 \mathbf{\Gamma M}(\boldsymbol{\lambda}) \mathbf{\Gamma}^T \mathbf{D}_2$, we have

$$\gamma_{\min}\left(\mathbf{\Gamma M}(\boldsymbol{\lambda})\mathbf{\Gamma}^T\right) \frac{4(1-\cos\frac{i\pi}{n})^2}{(2+\cos\frac{i\pi}{n})^2} \leq \tau_i \leq \gamma_{\max}\left(\mathbf{\Gamma M}(\boldsymbol{\lambda})\mathbf{\Gamma}^T\right) \frac{4(1-\cos\frac{i\pi}{n})^2}{(2+\cos\frac{i\pi}{n})^2},$$

where $\gamma_{\min}\left(\mathbf{\Gamma M}(\boldsymbol{\lambda})\mathbf{\Gamma}^T\right)$ and $\gamma_{\max}\left(\mathbf{\Gamma M}(\boldsymbol{\lambda})\mathbf{\Gamma}^T\right)$ are the minimum and maximum eigenvalues of $\mathbf{\Gamma M}(\boldsymbol{\lambda})\mathbf{\Gamma}^T$. Recall $\mathbf{\Gamma}$ is orthogonal, so we have $\gamma_{\min}\left(\mathbf{\Gamma M}(\boldsymbol{\lambda})\mathbf{\Gamma}^T\right) = \gamma_{\min}(\mathbf{M}(\boldsymbol{\lambda}))$ and $\gamma_{\max}\left(\mathbf{\Gamma M}(\boldsymbol{\lambda})\mathbf{\Gamma}^T\right) = \gamma_{\max}(\mathbf{M}(\boldsymbol{\lambda}))$. We will need the following lemma to examine the coefficient of variations condition.

Lemma A.4.2 *If there are two functions* $0 < C_1(n) < C_2(n)$ *such that*

$$\frac{C_2(n)}{C_1(n)} < c \text{ where } c > 0 \text{ is a constant,}$$

and

$$C_1(n) \cdot \frac{\left(1-\cos\frac{i\pi}{n}\right)^2}{(2+\cos\frac{i\pi}{n})^2} \leq \tau_i \leq C_2(n) \cdot \frac{\left(1-\cos\frac{i\pi}{n}\right)^2}{(2+\cos\frac{i\pi}{n})^2},$$

then we have, for any m satisfying $\frac{m}{n} \to 0$,

$$\frac{\left(\sum_m^n \tau_i^{-1}/n\right)^2}{\sum_m^n \tau_i^{-2}/n} \to 0 \quad as \quad n \to \infty.$$

Proof. For the numerator, we have

$$\begin{aligned} & \frac{1}{n}\sum_m^n \frac{(2+\cos\frac{i\pi}{n})^2}{\left(1-\cos\frac{i\pi}{n}\right)^2} \le \frac{1}{n}\sum_m^n \frac{9}{\left(1-\cos\frac{i\pi}{n}\right)^2} \approx \int_{\frac{m}{n}}^1 \frac{9}{(1-\cos x\pi)^2}dx \\ = & -\frac{(-2+\cos\frac{m\pi}{n})\cot\frac{m\pi}{2n}(\csc\frac{m\pi}{2n})^2}{2\pi} = A_1. \end{aligned}$$

For the denominator, we have

$$\begin{aligned} & \frac{1}{n}\sum_m^n \frac{\left(2+\cos\frac{i\pi}{n}\right)^4}{\left(1-\cos\frac{i\pi}{n}\right)^4} \ge \frac{1}{n}\sum_m^n \frac{1}{\left(1-\cos\frac{i\pi}{n}\right)^4} \approx \int_{\frac{m}{n}}^1 \frac{1}{(1-\cos x\pi)^4}dx \\ = & -\frac{(-32+29\cos\frac{m\pi}{n}-8\cos\frac{2m\pi}{n}+\cos\frac{3m\pi}{n})\cot\frac{m\pi}{2n}(\csc\frac{m\pi}{2n})^6}{560\pi} = A_2. \end{aligned}$$

Now we put everything together:

$$\frac{\left(\sum_m^n \frac{1}{\tau_i}/n\right)^2}{\sum_m^n \frac{1}{\tau_i^2}/n} \le \frac{A_1^2 C_1^{-2}(n)}{A_2 C_2^{-2}(n)} = -\frac{C_2^2(n)}{C_1^2(n)} \frac{420\cos\frac{m\pi}{2n}(-2+\cos\frac{m\pi}{n})^2\sin\frac{m\pi}{2n}}{\pi(-32+29\cos\frac{m\pi}{n}-8\cos\frac{2m\pi}{n}+\cos\frac{3m\pi}{n})} \to 0.$$

□

Combining the discussion after Lemma A.4.1 and Lemma A.4.2, if we can find a sufficient condition of $\mathbf{M}(\boldsymbol{\lambda})$ such that $\gamma_{\max}(\mathbf{M}(\boldsymbol{\lambda}))/\gamma_{\min}(\mathbf{M}(\boldsymbol{\lambda}))$ is upper bounded by a constant, then we find a sufficient condition for **(A.2)**. We have the following lemma.

Lemma A.4.3 *Let* $\lambda_{\min}$ *and* $\lambda_{\max}$ *denote the minimum and maximum among* $\lambda_k' s$, $1 \le k \le n-1$. *We have*

$$\frac{\gamma_{\max}(\mathbf{M}(\boldsymbol{\lambda}))}{\gamma_{\min}(\mathbf{M}(\boldsymbol{\lambda}))} \le 3 \cdot \frac{\lambda_{\max}}{\lambda_{\min}}.$$

Proof. Let $\mathbf{M}(\boldsymbol{\lambda})_{ij}$ be the (i,j) entry of $\mathbf{M}(\boldsymbol{\lambda})$, and γ denote an eigenvalue of $\mathbf{M}(\boldsymbol{\lambda})$. Due to the Gershgorin circle theorem ([24]), we have $|\gamma - \mathbf{M}(\boldsymbol{\lambda})_{ii}| \le \sum_{j\neq i} |\mathbf{M}(\boldsymbol{\lambda})_{ij}|$, $1 \le i \le n-2$. Consequently, we have

$$|\gamma| \ge |\mathbf{M}(\boldsymbol{\lambda})_{ii}| - |\gamma - \mathbf{M}(\boldsymbol{\lambda})_{ii}| \ge |\mathbf{M}(\boldsymbol{\lambda})_{ii}| - \sum_{j\neq i} |\mathbf{M}(\boldsymbol{\lambda})_{ij}|. \tag{4.70}$$

By recalling the structure of $\mathbf{M}(\boldsymbol{\lambda})$, we have that

$$|\mathbf{M}(\boldsymbol{\lambda})_{11}| - \sum_{j\neq 1} |\mathbf{M}(\boldsymbol{\lambda})_{1j}| = \lambda_1 + \frac{1}{2}\lambda_2, \tag{4.71}$$

$$|\mathbf{M}(\boldsymbol{\lambda})_{n-2,n-2}| - \sum_{j\neq n-2} |\mathbf{M}(\boldsymbol{\lambda})_{n-2,j}| = \frac{1}{2}\lambda_{n-2} + \lambda_{n-1}, \text{ and} \tag{4.72}$$

$$|\mathbf{M}(\boldsymbol{\lambda})_{ii}| - \sum_{j\neq i} |\mathbf{M}(\boldsymbol{\lambda})_{ij}| = \frac{1}{2}\lambda_i + \frac{1}{2}\lambda_{i+1}, \quad 2 \leq i \leq n-3. \tag{4.73}$$

Combining (4.70) through (4.73), we have $\gamma_{\min}(\mathbf{M}(\boldsymbol{\lambda})) \geq \lambda_{\min}$. On the other hand, we have $|\gamma| - |\mathbf{M}(\boldsymbol{\lambda})_{ii}| \leq |\gamma - \mathbf{M}(\boldsymbol{\lambda})_{ii}| \leq \sum_{j\neq i} |\mathbf{M}(\boldsymbol{\lambda})_{ij}|$. Consequently, we have

$$|\gamma| \leq |\mathbf{M}(\boldsymbol{\lambda})_{ii}| + \sum_{j\neq i} |\mathbf{M}(\boldsymbol{\lambda})_{ij}|. \tag{4.74}$$

On the other hand, we have

$$|\mathbf{M}(\boldsymbol{\lambda})_{11}| + \sum_{j\neq 1} |\mathbf{M}(\boldsymbol{\lambda})_{1j}| = \lambda_1 + \frac{3}{2}\lambda_2, \tag{4.75}$$

$$|\mathbf{M}(\boldsymbol{\lambda})_{n-2,n-2}| + \sum_{j\neq n-2} |\mathbf{M}(\boldsymbol{\lambda})_{n-2,j}| = \frac{3}{2}\lambda_{n-2} + \lambda_{n-1}, \text{ and} \tag{4.76}$$

$$|\mathbf{M}(\boldsymbol{\lambda})_{ii}| + \sum_{j\neq i} |\mathbf{M}(\boldsymbol{\lambda})_{ij}| = \frac{3}{2}\lambda_i + \frac{3}{2}\lambda_{i+1}, \quad 2 \leq i \leq n-3. \tag{4.77}$$

From (4.74) through (4.77), we have $\gamma_{\max}(\mathbf{M}(\boldsymbol{\lambda})) \leq 3\lambda_{\max}$. From all the above, we prove the lemma. □

The above lemma demonstrates that if we have $\lambda_{\max}/\lambda_{\min}$ upper bounded by a constant, then the **(A.2)** type condition is true for the terms on both sides of (4.69). Note they are the $(n-2) \times (n-2)$ principal submatrix of $\boldsymbol{\Sigma}(\boldsymbol{\lambda})$. The relation of the eigenvalues of a principal submatrix and the eigenvalues of the original matrix is known due to the interlacing theorem by Cauchy [45]. Cauchy's interlacing theorem states that if we let τ_i' be the eigenvalues of $\boldsymbol{\Sigma}(\boldsymbol{\lambda})$, then $\tau_i' \leq \tau_i \leq \tau_{i+1}'$. In essence, they have nearly identical behavior. It is not hard to show that when the condition in Lemma A.4.3 is met, the condition **(A.2)** is satisfied for $\boldsymbol{\Sigma}(\boldsymbol{\lambda})$ as well. We skip some details and claim that Theorem 1.2.7 is established.

A.5 Proof of Theorem 1.2.9

Recall the Stein estimates ($\tilde{\mathbf{f}}_n$), the associated unbiased risk estimate ($SURE_n$), and the true loss ($\tilde{L}_n$) defined in Theorem 1.2.5. Note in Theorem 1.2.5 that $SURE_n(\boldsymbol{\lambda})$ is a uniformly consistent estimate of $\tilde{L}_n(\boldsymbol{\lambda})$ over $\mathbf{f}_n$ and $\boldsymbol{\lambda}$. Also note that by comparing (2.14) and (2.15), $\hat{\boldsymbol{\lambda}}_{mG}$ also minimizes $SURE_n(\boldsymbol{\lambda})$. We first need the following Lemmas A.5.1 and A.5.2 to establish the upcoming Lemma A.5.3.

Lemma A.5.1 *Under **(A.3)**, we have $\tilde{L}_n(\boldsymbol{\lambda}_n) \to 0$, where $\boldsymbol{\lambda}_n$ is from **(A.3)**.*

Proof. We have

$$\begin{aligned}
\tilde{L}_n(\boldsymbol{\lambda}_n) &= \frac{1}{n}\|\tilde{\mathbf{f}}_n(\boldsymbol{\lambda}_n) - \mathbf{f}_n\|^2 = \frac{1}{n}\left\|\boldsymbol{\epsilon}_n - \sigma^2\frac{\mathrm{tr}(\mathbf{I}-\mathbf{S}_n(\boldsymbol{\lambda}_n))}{\|(\mathbf{I}-\mathbf{S}_n(\boldsymbol{\lambda}_n))\mathbf{y}_n\|^2}(\mathbf{I}-\mathbf{S}_n(\boldsymbol{\lambda}_n))\mathbf{y}_n\right\|^2 \\
&\leq \frac{1}{n}\left(1 - \frac{\sigma^2\mathrm{tr}(\mathbf{I}-\mathbf{S}_n(\boldsymbol{\lambda}_n))}{\|(\mathbf{I}-\mathbf{S}_n(\boldsymbol{\lambda}_n))\mathbf{y}_n\|^2}\right)^2\|\boldsymbol{\epsilon}_n\|^2 \\
&\quad + \frac{2}{n}\left|1 - \frac{\sigma^2\mathrm{tr}(\mathbf{I}-\mathbf{S}_n(\boldsymbol{\lambda}_n))}{\|(\mathbf{I}-\mathbf{S}_n(\boldsymbol{\lambda}_n))\mathbf{y}_n\|^2}\right|\left|\frac{\sigma^2\mathrm{tr}(\mathbf{I}-\mathbf{S}_n(\boldsymbol{\lambda}_n))}{\|(\mathbf{I}-\mathbf{S}_n(\boldsymbol{\lambda}_n))\mathbf{y}_n\|^2}\right|\|\boldsymbol{\epsilon}_n\|\cdot\|\mathbf{f}_n - \mathbf{S}_n(\boldsymbol{\lambda}_n)\mathbf{y}\| \\
&\quad + \frac{1}{n}\left(\sigma^2\frac{\mathrm{tr}(\mathbf{I}-\mathbf{S}_n(\boldsymbol{\lambda}_n))}{\|(\mathbf{I}-\mathbf{S}_n(\boldsymbol{\lambda}_n))\mathbf{y}_n\|^2}\right)^2\|\mathbf{f}_n - \mathbf{S}_n(\boldsymbol{\lambda}_n)\mathbf{y}\|^2.
\end{aligned}$$

It suffices to show that

$$\frac{\|(\mathbf{I}-\mathbf{S}_n(\boldsymbol{\lambda}_n))\mathbf{y}_n\|^2}{\mathrm{tr}(\mathbf{I}-\mathbf{S}_n(\boldsymbol{\lambda}_n))} \to \sigma^2. \tag{5.78}$$

Note that from the fact $\sigma^2(n^{-1}\mathrm{tr}\mathbf{S}_n(\boldsymbol{\lambda}_n))^2 \leq \sigma^2 n^{-1}\mathrm{tr}\mathbf{S}_n^2(\boldsymbol{\lambda}_n) \leq \mathbb{E}L_n(\boldsymbol{\lambda}_n) \to 0$, we have $n^{-1}\mathrm{tr}\mathbf{S}_n(\boldsymbol{\lambda}_n) \to 0$. Thus, in the denominator of (5.78), we have

$$n^{-1}\mathrm{tr}(\mathbf{I}-\mathbf{S}_n(\boldsymbol{\lambda}_n)) = 1 - n^{-1}\mathrm{tr}\mathbf{S}_n(\boldsymbol{\lambda}_n) \to 1. \tag{5.79}$$

We also have, for the numerator of (5.78),

$$n^{-1}\|(\mathbf{I}-\mathbf{S}_n(\boldsymbol{\lambda}_n))\mathbf{y}_n\|^2 \leq n^{-1}\|\boldsymbol{\epsilon}_n\| + n^{-1}\|\mathbf{f}_n - \hat{\mathbf{f}}_n\|^2 + 2n^{-1}|\langle\boldsymbol{\epsilon}_n, \mathbf{f}_n - \hat{\mathbf{f}}_n\rangle| \to \sigma^2, \tag{5.80}$$

by the fact $n^{-1}\|\boldsymbol{\epsilon}_n\|^2 \to \sigma^2$, **(A.3)**, and the Cauchy-Schwartz inequality. Finally, (5.78) follows from (5.79) and (5.80). □

Lemma A.5.2 *Under **(A.3)**, we have* $\tilde{L}_n(\hat{\boldsymbol{\lambda}}_{mG}) \to 0$.

Proof. From the uniform consistency of $SURE_n(\boldsymbol{\lambda})$, together with the fact that $\hat{\boldsymbol{\lambda}}_{mG}$ also minimizes $SURE_n(\boldsymbol{\lambda})$, we have $\tilde{L}_n(\hat{\boldsymbol{\lambda}}_{mG}) = SURE_n(\hat{\boldsymbol{\lambda}}_{mG}) + o_p(1) \leq SURE_n(\boldsymbol{\lambda}_n) + o_p(1) = \tilde{L}_n(\boldsymbol{\lambda}_n) + o_p(1) = o_p(1)$, where the last equality follows from Lemma A.5.1. □

Once again using the fact of the uniform consistency of $SURE_n(\boldsymbol{\lambda})$, from the result of Lemma A.5.2, we have $SURE_n(\hat{\boldsymbol{\lambda}}_{mG}) \to 0$. Equivalently, we have the following Lemma.

Lemma A.5.3 *Under **(A.3)**, we have* $mGCV_n(\hat{\boldsymbol{\lambda}}_{mG}) \to \sigma^2$.

We also need the following Lammas A.5.4, A.5.5 and A.5.6.

Lemma A.5.4 *If* $\epsilon_i s$ *are i.i.d.* $N(0, \sigma^2)$,

$$\lim_{n\to\infty} P\left\{ \frac{\|(\mathbf{I} - \mathbf{S}_n(\hat{\boldsymbol{\lambda}}))\mathbf{y}_n\|^2}{\|(\mathbf{I} - \mathbf{S}_n(\hat{\boldsymbol{\lambda}}))\mathbf{f}_n\|^2 + \sigma^2 tr(\mathbf{I} - \mathbf{S}_n(\hat{\boldsymbol{\lambda}}))^2} \leq 1 - \delta \right\} = 0, \text{ for any } \delta > 0. \quad (5.81)$$

Proof. It can be proved similarly as in [30]: See the proof of Lemma 5.2 in [30][pp.1374-1376]. The key is to upper bound five terms in (7.3.8) in [30] with a small quantity $\epsilon/5$. Note that our smoothing matrix $\mathbf{S}(\boldsymbol{\lambda})$ has the same canonical form of (4.9) in [30] with λ_i replaced by τ_i. A careful check of the proof in [30] reveals that the same argument applies for arbitrary λ_i, hence the above lemma can be established accordingly. □

Lemma A.5.5 *For any sequence* $\hat{\boldsymbol{\lambda}}$ *such that*

$$mGCV_n(\hat{\boldsymbol{\lambda}}) \to \sigma^2, \quad (5.82)$$

*under **(A.2)**, we have* $n^{-1} tr\mathbf{S}_n(\hat{\boldsymbol{\lambda}}) \to 0$.

Proof. Using (5.82) and (5.81), we have $[n^{-1}\mathrm{tr}(\mathbf{I}-\mathbf{S}_n(\hat{\boldsymbol{\lambda}}))]^2 \geq [n^{-1}\mathrm{tr}(\mathbf{I}-\mathbf{S}_n(\hat{\boldsymbol{\lambda}}))^2](1-o_p(1))$. Then with the fact that $[n^{-1}\mathrm{tr}(\mathbf{I}-\mathbf{S}_n(\hat{\boldsymbol{\lambda}}))]^2 \leq n^{-1}\mathrm{tr}(\mathbf{I}-\mathbf{S}_n(\hat{\boldsymbol{\lambda}}))^2$, we have the following:

$$\frac{[n^{-1}\mathrm{tr}(\mathbf{I}-\mathbf{S}_n(\hat{\boldsymbol{\lambda}}))]^2}{n^{-1}\mathrm{tr}(\mathbf{I}-\mathbf{S}_n(\hat{\boldsymbol{\lambda}}))^2} \to 1. \tag{5.83}$$

Recall $\mathbf{S}_n(\hat{\boldsymbol{\lambda}}) = \left(\mathbf{I}+\boldsymbol{\Sigma}_n(\hat{\boldsymbol{\lambda}})\right)^{-1}$, where $\boldsymbol{\Sigma}_n(\hat{\boldsymbol{\lambda}}) = \frac{h}{3}\mathbf{Q}^T\mathbf{M}^{-1}\mathbf{M}(\hat{\boldsymbol{\lambda}})\mathbf{M}^{-1}\mathbf{Q}$, and the eigenvalues of $\boldsymbol{\Sigma}_n(\hat{\boldsymbol{\lambda}})$ are $0 \leq \tau_1 \leq \tau_2 \leq \cdots \leq \tau_n$. It is clear that the eigenvalues of $\mathbf{I}-\mathbf{S}_n(\hat{\boldsymbol{\lambda}})$ are $\tau_i(1+\tau_i)^{-1}$. Similarly as in [30], let τ be the random variable taking values τ_i with probability n^{-1} for each i. Then (5.83) means $(\mathbb{E}\tau(1+\tau)^{-1})^2/\mathbb{E}\tau^2(1+\tau)^{-2} \to 1$, which implies

$$\tau(1+\tau)^{-1}/\mathbb{E}\tau(1+\tau)^{-1} \to 1 \text{ in probability.} \tag{5.84}$$

Since (5.84) implies that both $\tau_{[pn]}(1+\tau_{[pn]})^{-1}$ and $\tau_{[qn]}(1+\tau_{[qn]})^{-1}$ tend to $\mathbb{E}\tau(1+\tau)^{-1}$, we have $\mathbb{E}\tau(1+\tau)^{-1} \to 1$ due to **(A.2′)**. It is clear that $\mathbb{E}\tau(1+\tau)^{-1} \to 1$ implies $n^{-1}\mathrm{tr}\mathbf{S}_n(\hat{\boldsymbol{\lambda}}) \to 0$. □

Lemma A.5.6 *For any sequence $\hat{\boldsymbol{\lambda}}$ such that $mGCV_n(\hat{\boldsymbol{\lambda}}) \to \sigma^2$, $\hat{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}})$ is consistent if and only if $n^{-1}tr\mathbf{S}_n(\hat{\boldsymbol{\lambda}}) \to 0$.*

Proof. If $\hat{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}})$ is consistent, $L_n(\hat{\boldsymbol{\lambda}}) \to 0$, and hence, $n^{-1}\|\mathbf{y}_n - \hat{\mathbf{f}}_n(\boldsymbol{\lambda})\|^2 \to \sigma^2$, since $n^{-1}\|\boldsymbol{\epsilon}_n\| \to \sigma^2$. Then, from the fact that

$$\mathrm{mGCV}_n(\hat{\boldsymbol{\lambda}}) = n^{-1}\|\mathbf{y}_n - \hat{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}})\|_2^2/[n^{-1}\mathrm{tr}(\mathbf{I}-\mathbf{S}_n(\hat{\boldsymbol{\lambda}}))]^2 \to \sigma^2,$$

we have $[n^{-1}\mathrm{tr}(\mathbf{I}-\mathbf{S}_n(\hat{\boldsymbol{\lambda}}))]^2 \to 1$, and thus, $n^{-1}\mathrm{tr}\mathbf{S}_n(\hat{\boldsymbol{\lambda}}) \to 0$. Conversely, if $n^{-1}\mathrm{tr}\mathbf{S}_n(\hat{\boldsymbol{\lambda}}) \to 0$, since $mGCV_n(\hat{\boldsymbol{\lambda}}) \to \sigma^2$, we have $n^{-1}\|\mathbf{y}_n - \hat{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}})\|^2 \to \sigma^2$. Then, with the fact that $n^{-1}\|\boldsymbol{\epsilon}_n\| \to \sigma^2$, we have $L_n(\hat{\boldsymbol{\lambda}}) \to 0$, which implies $\hat{\mathbf{f}}_n(\hat{\boldsymbol{\lambda}})$ is consistent. □

From Lemmas A.5.3, A.5.5, and A.5.6, Theorem 1.2.9 is proved.

A.6 Proof of Theorem 1.2.11

For the proof, we use Theorem 1.2.6. Note that in Theorem 1.2.6, besides **(A.1)**, we need two conditions: (2.16) and (2.17). Under the conditions **(A.2)** and **(A.3)**, (2.16) is true with $\hat{\boldsymbol{\lambda}}_{mG}$ due to Theorem 1.2.9. The following lemma states that under **(A.1)** and **(A.2)**, (2.16) implies (2.17); therefore we prove Theorem 1.2.11.

Lemma A.6.1 *Under **(A.1)** and **(A.2)**, for any $\boldsymbol{\lambda}$ such that*

$$L_n(\boldsymbol{\lambda}) \to 0, \tag{6.85}$$

we have

$$\frac{(n^{-1}tr\mathbf{S}_n(\boldsymbol{\lambda}))^2}{n^{-1}tr\mathbf{S}_n^2(\boldsymbol{\lambda})} \to 0, \tag{6.86}$$

Proof. Recall $\mathbf{S}_n(\boldsymbol{\lambda}) = (\mathbf{I} + \boldsymbol{\Sigma}_n(\boldsymbol{\lambda}))^{-1}$, where $\boldsymbol{\Sigma}_n(\boldsymbol{\lambda}) = \frac{h}{3}\mathbf{Q}^T\mathbf{M}^{-1}\mathbf{M}(\boldsymbol{\lambda})\mathbf{M}^{-1}\mathbf{Q}$, and the eigenvalues of $\boldsymbol{\Sigma}_n(\boldsymbol{\lambda})$ are $0 \le \tau_1 \le \tau_2 \le \cdots \le \tau_n$. We have

$$\frac{(n^{-1}\mathrm{tr}\mathbf{S}_n(\boldsymbol{\lambda}))^2}{n^{-1}\mathrm{tr}\mathbf{S}_n^2(\boldsymbol{\lambda})} = \frac{\left(n^{-1}\sum_{i=1}^n(1+\tau_i)^{-1}\right)^2}{n^{-1}\sum_{i=1}^n(1+\tau_i)^{-2}}.$$

Define $m = i$ such that $\tau_i \le 1 \le \tau_{i+1}$. Then we have

$$\sum_{i=1}^n(1+\tau_i)^{-1} \le m + \sum_{i=m+1}^n \tau_i^{-1},$$

and

$$\sum_{i=1}^n(1+\tau_i)^{-2} \ge \frac{1}{4}\left(m + \sum_{i=m+1}^n \tau_i^{-2}\right). \tag{6.87}$$

To show (6.86), therefore, it suffices to show

$$\frac{\left(\frac{m}{n} + \frac{1}{n}\sum_{i=m+1}^n \tau_i^{-1}\right)^2}{\frac{m}{n} + \frac{1}{n}\sum_{i=m+1}^n \tau_i^{-2}} \to 0. \tag{6.88}$$

On the other hand, from (6.85), we have $\mathbb{E}L_n(\boldsymbol{\lambda}) \to 0$ due to (3.59), and hence $n^{-1}\mathrm{tr}\mathbf{S}_n^2(\boldsymbol{\lambda}) \to 0$. Thus $m/n \to 0$ due to (6.87). Bringing this to (6.88), we have (6.86) under **(A.2)**. □

A.7 Proof of Theorem 1.2.12

It is clear that (6.85) is equivalent to (2.16). Then if we have any $\hat{\boldsymbol{\lambda}}$ such that $L_n(\hat{\boldsymbol{\lambda}}) \to 0$, by Lemma A.6.1, we can utilize Theorem 1.2.6. Under **(A.3)**, this holds for $\hat{\boldsymbol{\lambda}} = \boldsymbol{\lambda}_n^*$, where $\boldsymbol{\lambda}_n^*$ is the minimizer of $L_n(\boldsymbol{\lambda})$. Thus we have

$$SURE_n(\boldsymbol{\lambda}_n^*) - n^{-1}\|\boldsymbol{\epsilon}_n\|^2 + \sigma^2 = L_n(\boldsymbol{\lambda}_n^*)(1 + o_p(1)).$$

On the other hand, by Theorem 1.2.9, this also holds for $\hat{\boldsymbol{\lambda}} = \hat{\boldsymbol{\lambda}}_{mG}$. Therefore we have

$$SURE_n(\hat{\boldsymbol{\lambda}}_{mG}) - n^{-1}\|\boldsymbol{\epsilon}_n\|^2 + \sigma^2 = L_n(\hat{\boldsymbol{\lambda}}_{mG})(1 + o_p(1)).$$

Since $SURE_n(\hat{\boldsymbol{\lambda}}_{mG}) \leq SURE_n(\boldsymbol{\lambda}_n^*)$ and $L_n(\boldsymbol{\lambda}_n^*) \leq L_n(\hat{\boldsymbol{\lambda}}_{mG})$, we have $L_n(\hat{\boldsymbol{\lambda}}_{mG})/L_n(\boldsymbol{\lambda}_n^*) \to 1$.

A.8 Proof of Corollary 1.2.13

If we replace **(A.3)** by **(A.3′)**, and replace $\hat{\boldsymbol{\lambda}}$ and $\boldsymbol{\lambda}_n^*$ by $\hat{\boldsymbol{\lambda}}_\Omega$ and $\boldsymbol{\lambda}_{n,\Omega}^*$, respectively, where $\hat{\boldsymbol{\lambda}}_\Omega \in \Omega$ satisfies $L_n(\hat{\boldsymbol{\lambda}}_\Omega) \to 0$ and $\boldsymbol{\lambda}_{n,\Omega}^*$ is the minimizer of $L_n(\boldsymbol{\lambda})$ over Ω, we can see that the same argument as in the proof of Theorem 1.2.12 works.

A.9 Proof of Theorem 1.3.3

Let $\sigma_i = f^{(2)}(x_i), i = 2, \cdots, n-1$, $\sigma_1 = \sigma_n = 0$, and $h = x_{i+1} - x_i, 1 \leq i \leq n-1$. We can easily verify the following with standard calculation:

$$\int_{x_i}^{x_{i+1}} f^{(2)}(x)^2 dx = \frac{h}{3}\sigma_i^2 + \frac{h}{3}\sigma_{i+1}^2 + \frac{h}{3}\sigma_i\sigma_{i+1}, \quad i = 1, \cdots, n-1. \tag{9.89}$$

With the approximation that $\lambda(x) = \lambda_i$ for $x \in [x_i, x_{i+1}]$, using (9.89), the objective function in (3.21) can be restated as

$$\frac{1}{n}(\mathbf{y} - \mathbf{f})^T(\mathbf{y} - \mathbf{f}) + \frac{h}{3}\sum_{i=1}^{n-1}\lambda_i(\sigma_i^2 + \sigma_{i+1}^2 + \sigma_i\sigma_{i+1}). \tag{9.90}$$

For the penalty term in (9.90), by introducing $\mathbf{M}(\boldsymbol{\lambda})$, it can be shown that

$$\sum_{i=1}^{n-1}\lambda_i(\sigma_i^2 + \sigma_{i+1}^2 + \sigma_i\sigma_{i+1}) = \sigma^T\mathbf{M}(\boldsymbol{\lambda})\sigma. \tag{9.91}$$

Combining (9.90) and (9.91), the objective function in (3.21) is rewritten as

$$\frac{1}{n}(\mathbf{y}-\mathbf{f})^T(\mathbf{y}-\mathbf{f})+\frac{h}{3}\sigma^T\mathbf{M}(\boldsymbol{\lambda})\sigma. \tag{9.92}$$

To find the minimizer of (9.92), using the fact that $\mathbf{M}\sigma = \mathbf{Qf}$ and considering the first order condition, we have the closed-form solution in (3.33). □

A.10 Proof of Lemma 2.3.1

Proof. Writing the condition in (3.47) in a generic form, we have

$$\min \|\alpha\|_2^2, \qquad \text{subject to } A\alpha = b,$$

where A is a matrix of full row rank and b is a vector. We consider its Lagrangian:

$$L(\alpha,\lambda)=\alpha^T\alpha+\sum_{i=1}^{r}\lambda_i(A_i\alpha-b_i)=\alpha^T\alpha+\lambda^T(A\alpha-b),$$

where λ_i is the ith Lagrangian multiplier, which is also the ith component of vector λ, A_i is the ith row of A, and b_i is the ith entry of b. The above achieves minimum if and only if

$$\begin{cases} 2\alpha + A^T\lambda = 0, \\ A\alpha = b. \end{cases}$$

The above can be rewritten as

$$\begin{pmatrix} 2I & A^T \\ A & 0 \end{pmatrix}\begin{pmatrix} \alpha \\ \lambda \end{pmatrix}=\begin{pmatrix} 0 \\ b \end{pmatrix}.$$

Hence we have

$$\begin{pmatrix} \alpha \\ \lambda \end{pmatrix}=\begin{pmatrix} 2I & A^T \\ A & 0 \end{pmatrix}^{-1}\begin{pmatrix} 0 \\ b \end{pmatrix}.$$

Notice that we have

$$\begin{pmatrix} 2I & A^T \\ A & 0 \end{pmatrix}^{-1}=\begin{pmatrix} \frac{1}{2}I-\frac{1}{2}A^T(AA^T)^{-1}A & A^T(AA^T)^{-1} \\ (AA^T)^{-1}A & -2(AA^T)^{-1} \end{pmatrix},$$

which brings us the following closed-form solution

$$\begin{pmatrix} \alpha \\ \lambda \end{pmatrix} = \begin{pmatrix} A^T(AA^T)^{-1}b \\ -2(AA^T)^{-1}b \end{pmatrix}.$$

For our purpose, we have $A = \Phi_1$ and $b = y - \Phi_2\beta$, which leads to (3.48). □

A.11 Determination of ν in Section 2.3.3

We consider how to determine ν, such that the resulting estimator is unbiased. Based on our statistical model, ν depends on three sets of parameters: (1) A and σ reflecting the smoothness of the surface, (2) ϵ, standard deviation of the measurement errors (same notation as in step 3 in Section 2.5.2), and (3) α, coefficients of the scaling functions, reflecting the shape. It is hard to derive the value of ν analytically. We use simulations instead. A and σ are estimable via scanning data; corresponding strategy has been discussed in Section 2.4.1. For ϵ, one can first take the differences of neighboring CMM measurements, then consider the median absolute value (MAD) statistic as an estimator. If straightness is the target, we can simply take $\alpha = 0$.

When A, σ, ϵ, and α are given, simulations can be used to choose ν. Let N_1 denote the total number of random y's that are generated (similar to the Step 1 in Section 2.5.2). Let N_2 denote the number of surrogates (whose generation is described in Step 2 in Section 2.3.3). Furthermore, let e_{ij} denote the form error of the jth surrogate in the i-th experiment, $i = 1, \ldots, N_1, j = 1, \ldots, N_2$. Let e_{i0} denote the computed form error for the y in the i-th experiment. Without loss of generality, we assume that for fixed i, e_{ij}'s are nondecreasing as j increases. We find the j that solves

$$\min_{j:1\le j\le N_2} \quad \sum_{i=1}^{N_1} \left| \frac{e_{ij}}{e_{i0}} - 1 \right|.$$

The above is to find the empirical quantile, such that the corresponding e_{ij}'s are closest to the e_{i0}'s. The j^*/N_2 (with j^* being the minimizer) is the desirable value of ν. In our experiments in Section 2.5.2, by using the estimated parameters $\hat{A} = 1.0444$,

$\hat{\sigma} = 0.9720$, $\hat{\varepsilon} = 0.2013$, and setting $\alpha = 0$, we found that $\nu = 0.974$, which is used in our simulations.

A.12 Proof of Theorem 3.2.1

Proof. We show how an $O(N)$ algorithm can be constructed. First of all, the partial sums:

$$p_i = \sum_{x=1}^{i} \frac{\hat{b}_1(x) - \hat{b}_2(x)}{\sigma(x)}, \quad 1 \leq i \leq N,$$

can be done in $O(N)$, because we have

$$p_{i+1} = p_i + \frac{\hat{b}_1(i+1) - \hat{b}_2(i+1)}{\sigma(i+1)}, \quad 1 \leq i \leq N-1.$$

Secondly, differences $p_{i+s-1} - p_i, 1 \leq i \leq N-s+1$ can be computed at $O(N)$. Evidently, taking the maximum among these differences can be done at $O(N)$. The above solves the following

$$\max_{1 \leq i \leq N-s+1} \sum_{x=i}^{i+s-1} \frac{\hat{b}_1(x) - \hat{b}_2(x)}{\sigma(x)}.$$

Note that comparing to (2.51), there is no absolute value in the above objective function. To solve (2.51), we rerun the above for

$$\max_{1 \leq i \leq N-s+1} \sum_{x=i}^{i+s-1} \frac{\hat{b}_2(x) - \hat{b}_1(x)}{\sigma(x)};$$

(Note the switch of positions of $\hat{b}_1$ and $\hat{b}_2$ in the objective function) and take the maximum. □

REFERENCES

[1] Abramovich, F. and Grinshtein, V., "Derivation of equivalent kernel for general spline smoothing: a systematic approach," *Bernoulli*, vol. 5, no. 2, pp. 359–379, 1999.

[2] Abramovich, F. and Steinberg, D. M., "Improved inference in nonparametric regression using L_k-smoothing splines," *Journal of Statistical Planning and Inference*, vol. 49, no. 3, pp. 327–341, 1996.

[3] ANSI Z540.3, *Requirements for the Calibration of Measuring and Test Equipment*, 2006.

[4] ASME B89.7.3.1, *For Decision Rules: Considering Measurement Uncertainty In Determining Conformance To Specifications*, 2001.

[5] ASME Standard B46.1, *Surface Texture, Surface Roughness, Waviness, and Lay*, 2002.

[6] Bosch, J. A., ed., *Coordinate measuring machines and systems*. Marcel Dekker, New York, 1995.

[7] Collins, C. E., Fay, E. B., Aguirre-Cruz, J. A., and Raman, S., "Alternate methods for sampling in coordinate metrology," *Proceedings of the Institution of Mechanical Engineers Part B-Journal of Engineering Manufacture*, vol. 221, pp. 1041–1052, June 2007.

[8] Courant, R. and Hilbert, D., *Methods of Mathematical Physics: Vol I*. New York: Interscience Publishers, 1953.

[9] Craven, P. and Wahba, G., "Smoothing noisy data with spline functions," *Numerische Mathematik*, vol. 31, no. 4, pp. 377–403, 1979.

[10] Cummins, D. J., Filloon, T. G., and Nychka, D., "Confidence intervals for nonparametric curve estimates: toward more uniform pointwise coverage," *Journal of the American Statistical Association*, vol. 96, no. 453, pp. 233–246, 2001.

[11] Davies, P. L. and Meise, M., "Approximating data with weighted smoothing splines," *Journal of Nonparametric Statistics*, vol. 20, no. 3, pp. 207–228, 2008.

[12] Ding, Y., Elsayed, E. A., Kumara, S., Lu, J. C., Niu, F., and Shi, J. J., "Distributed sensing for quality and productivity improvements," *IEEE Transactions on Automation Science and Engineering*, vol. 3, pp. 344–359, October 2006.

[13] Donoho, D. L. and Johnstone, I. M., "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[14] Dowling, M. M., Griffin, P. M., Tsui, K.-L., and Zhou, C., "Statistical issues in geometric tolerance verification using coordinate measuring machines," *Technometrics*, vol. 39, pp. 1–24, January 1997.

[15] Edgeworth, R. and Wilhelm, R. G., "Adaptive sampling for coordinate metrology," *Precision Engineering*, vol. 23, pp. 144–154, July 1999.

[16] Eubank, R. L., *Nonparametric Regression and Spline Smoothing*. New York: Marcel Dekker, 1999.

[17] Fan, K.-C. and Leu, M. C., "Intelligent planning of cad-directed inspection for coordinate measuring machines," *Computer Integrated Manufacturing Systems*, vol. 11, no. 1-2, pp. 43–51, 1998.

[18] Gilbert, R. C., Raman, S., Trafalis, T. B., Obeidat, S. M., and Aguirre-Cruz, J. A., "Mathematical foundations for form inspection and adaptive sampling," *Journal of Manufacturing Science and Engineering-Transactions of the ASME*, vol. 131, p. 041001, August 2009.

[19] Girard, D. A., "Asymptotic optimality of the fast randomized versions of GCV and c_l in ridge regression and regularization," *The Annals of Statistics*, vol. 19, no. 4, pp. 1950–1963, 1991.

[20] Grant, M., Boyd, S., and Ye, Y., "CVX: Matlab software for disciplined convex programming," tech. rep., Stanford University, Stanford, CA, December 2007. Version 1.1, Build 585, downloadable at http://www.stanford.edu/∼boyd/cvx/.

[21] Green, P. J. and Silverman, B. W., *Nonparametric regression and generalized linear models: a roughness penalty approach*, vol. 58 of *Monographs on statistics and applied probability*. New York, NY: Chapman & Hall, 1994.

[22] Groover, M. P., *Fundamentals of Modern Manufacturing: Materials, Processes, and Systems*. Hoboken, NJ: John Wiley & Sons, 1999.

[23] Hocken, R. J., Raja, J., and Babu, U., "Sampling issues in coordinate metrology," *Manufacturing Review*, pp. 282–294, 1993.

[24] Horn, R. and Johnson, C. R., *Matrix Analysis*. Cambridge, 1985.

[25] Huo, X., Kim, S. B., Tsui, K. L., and Wang, S., "FBP: A frontier-based tree pruning algorithm," *INFORMS Journal on Computing*, vol. 18, no. 4, pp. 494–505, 2006.

[26] Jin, J. and Shi, J., "Feature-preserving data compression of stamping tonnage information using wavelets," *Technometrics*, vol. 41, pp. 327–339, November 1999.

[27] Jin, J. and Shi, J., "Press tonnage signal decomposition and validation analysis for transfer or progressive die processes," *ASME Transactions, Journal of Manufacturing Science and Engineering*, vol. 127, no. 1, pp. 231–235, 2005.

[28] Land, S. R. and Friedman, J. H., "Variable fusion: a new method of adaptive signal regression," tech. rep., Department of Statistics, Stanford University, 1996.

[29] Lee, G., Mou, J., and Shen, Y., "Sampling strategy design for dimensional measurement of geometric features using coordinate measuring machine," *International Journal of Machine Tools and Manufacture*, vol. 37, pp. 917–934, July 1997.

[30] Li, K.-C., "From Stein's unbised risk estimates to the method of generalized cross validation," *The Annals of Statistics*, vol. 13, no. 4, pp. 1352–1377, 1985.

[31] Li, K.-C., "Asymptotic optimality of C_L and generalized cross-validation in ridge regression with application to spline smoothing," *The Annals of Statistics*, vol. 14, no. 3, pp. 1101–1112, 1986.

[32] Liu, C., Ding, Y., and Chen, Y., "Optimal coordinate sensor placements for estimating mean and variance components of variation sources," *IIE Transactions*, vol. 37, pp. 877–889, September 2005.

[33] Liu, Z. and Guo, W., "Data driven adaptive spline smoothing," *Statistica Sinica*, vol. 20, pp. 1143–1163, 2010.

[34] Mallat, S., *A wavelet tour of signal processing*. San Diego, CA: Academic Press, 1998.

[35] Mallows, C. L., "Some comments on c_p," *Technometrics*, vol. 15, pp. 661–675, 1973.

[36] Muller, H.-G. and Stadmuller, U., "Variable bandwidth kernel estimators of regression curves," *The Annals of Statistics*, vol. 15, no. 1, pp. 182–201, 1987.

[37] Nychka, D., "Bayesian confidence intervals for smoothing splines," *Journal of the American Statistical Association*, vol. 83, no. 404, pp. 1134–1143, 1988.

[38] Pintore, A., Speckman, P., and Holmes, C. C., "Spatially adaptive smoothing splines," *Biometrika*, vol. 93, no. 1, pp. 113–125, 2006.

[39] Prakasvudhisarn, C. and Raman, S., "Framework for cone feature measurement using coordinate measuring machines," *Journal of Manufacturing Science and Engineering*, vol. 126, pp. 169–177, February 2004.

[40] Ramsay, J. O. and Silverman, B. W., *Functional Data Analysis*. New York: Springer, 1996.

[41] Rivest, L. P., "Some linear models for estimating the motion of rigid bodies with applications to geometric quality assurance," *Journal of the American Statistical Association*, vol. 93, pp. 632–642, June 1998.

[42] Schmitz, T. L., Ziegert, J. C., Canning, J. S., and Zapata, R., "Case study: A comparison of error sources in high-speed milling," *Precision Engineering*, vol. 32, pp. 126–133, 2008.

[43] Silverman, B. W., "Spline smoothing: the equivalent variable kernel method," *The Annals of Statistics*, vol. 12, no. 3, pp. 898–916, 1984.

[44] Slocum, A., *Precision Machine Design*. Upper Saddle River, NJ: Prentice-Hall, 1992.

[45] Stewart, G. W., *Matrix Perturbation Theory*. Academic Press, 1990.

[46] Stone, C. J., "Optimal global rates of convergence for nonparametric regression," *The Annals of Statistics*, vol. 10, no. 4, pp. 1040–1053, 1982.

[47] Stone, M., "Cross-validation and multinomial prediction," *Biometrika*, vol. 61, pp. 509–515, 1974.

[48] Storlie, C. B., Bondell, H. D., and Reich, B. J., "A locally adaptive penalty for estimation of functions with varying roughness," *To appear in Journal of Computational and Graphical Statistics*, 2010.

[49] Tibshirani, R., "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[50] Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K., "Sparsity and smoothness via the fused Lasso," *Journal of the Royal Statistical Society, Series B*, vol. 67, no. 1, pp. 91–108, 2005.

[51] Tlusty, J., *Manufacturing Processes and Equipment*. Upper Saddle River, NJ: Prentice-Hall, 2000.

[52] Wahba, G., "Smoothing noisy data with spline functions," *Numerische Mathematik*, vol. 24, no. 5, pp. 383–393, 1975.

[53] Wahba, G., "Bayesian "confidence intervals" for the cross-validated smoothing spline," *Journal of the Royal Statistical Society, Series B*, vol. 45, no. 1, pp. 133–150, 1983.

[54] Wahba, G., "A comparison of GCV and GML for choosing the smoothing parameter in the generalized spline smoothing problem," *The Annals of Statistics*, vol. 13, no. 4, pp. 1378–1402, 1985.

[55] Whitehouse, D., *Surfaces and Their Measurement.* London, UK: Hermes Penton Science, 2002.

[56] Xia, H., Ding, Y., and Wang, J., "Gaussian process method for form error assessment using coordinate measurements," *IIE Transactions*, vol. 40, pp. 931–946, 2008.

[57] Zhang, G., Veale, R., Charlton, T., Borchardt, B., and Hocken, R., "Error compensation of coordinate measuring machines," *CIRP Annals*, vol. 34, pp. 445–448, 1985.

[58] Zheng, H. Q., Li, X. P., Wong, Y. S., and Nee, A. Y. C., "Theoretical modelling and simulation of cutting forces in face milling with cutter runout," *International Journal of Machine Tools and Manufacture*, vol. 39, pp. 2003–2018, December 1999.